

1N-07
145784
P.70

NASA Contractor Report CR-187127

Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems

Task IV - Advanced Fan Section Aerodynamic Analysis
Computer Program User's Manual

Andrew J. Crook and Robert A. Delaney
Allison Gas Turbine Division of General Motors
Indianapolis, Indiana

November 1992



Prepared for
Lewis Research Center
Under Contract NAS3-25270

(NASA-CR-187127) INVESTIGATION OF
ADVANCED COUNTERROTATION BLADE
CONFIGURATION CONCEPTS FOR HIGH
SPEED TURBOPROP SYSTEMS. TASK 4:
ADVANCED FAN SECTION AERODYNAMIC
ANALYSIS COMPUTER PROGRAM USER'S
MANUAL Report, Jul. 1991 - Jul.
1992 (General Motors Corp.) 70 p

N93-18702

Unclas

H1/07 0145784

Preface

This manual was prepared by Andrew J. Crook and Robert A. Delaney of the Allison Gas Turbine Division, General Motors Corporation, Indianapolis, IN. The work was performed under NASA Contract NAS3-25270 from July, 1991 to July, 1992. The modification of the flow analysis for multiple splitter fan/engine geometries was performed by Andrew J. Crook. The Allison program manager for this contract was Robert A. Delaney. The NASA program manager for this contract was Christopher J. Miller.

Acknowledgements

The authors would like to express their appreciation to the following personnel who contributed to this program:

Dr. Edward J. Hall for his suggestions concerning the solver and the many helpful contributions to this document,

Dr. Christopher J. Miller for his review of the program,

UNIX is a trademark of AT&T

IRIS is a trademark of Silicon Graphics, Inc.

TABLE OF CONTENTS

1. SUMMARY	1
2. INTRODUCTION	3
3. ADPAC-APES PROGRAM INITIALIZATION AND DEMONSTRATION	7
3.1 Introduction	7
3.2 Extracting the Source Files	7
3.3 Compiling the Source Code	8
3.4 Running the Distribution Demonstration Test Case	10
4. ADPAC-APES PROGRAM GENERAL DESCRIPTION	15
4.1 Parameter/Array Sizes	15
4.2 Input File Description - Basic Input Section	16
4.3 Input File Description - Blade Row File Name Section	24
4.4 Input File Description - Extra Input Parameter Section	26
4.5 Output File Description	31
4.6 Subroutine Description	35
5. ADPAC-APES SOLUTION TECHNIQUES	41

5.1	Single/Multiple Blade Row Procedure	41
5.2	Solution Startup Techniques	49
	REFERENCES	53
	APPENDIX A. <i>FULLPLOT</i>: <i>PostScript</i> X-Y Plotting Routine	55
	APPENDIX B. <i>ADPAC-APES</i> DISTRIBUTION LIST	65

1. SUMMARY

The purpose of this study is the development of a three-dimensional Euler/Navier-Stokes flow analysis for fan section/engine geometries containing multiple blade rows and multiple spanwise flow splitters. An existing procedure developed by Dr J. J. Adamczyk and associates [1] at the NASA Lewis Research Center was modified to accept multiple spanwise splitter geometries and simulate engine core conditions. The procedure was also modified to allow coarse parallelization of the solution algorithm. This document is a user's manual for the code, developed under Task IV of NASA Contract NAS3-25270.

The numerical solution is based upon a finite volume technique with a four stage Runge-Kutta time marching procedure. Numerical dissipation is used to gain solution stability but is reduced in viscous dominated flow regions. Local time stepping and implicit residual smoothing are used to increase the rate of convergence. Multiple blade row solutions are based upon the average-passage system of equations. The numerical solutions are performed on an H-type grid system, with meshes being generated by the system (TIGG3D) developed earlier under this contract. The grid generation scheme meets the average-passage requirement of maintaining a common axisymmetric mesh for each blade row grid.

The analysis was run on several geometry configurations ranging from one to five blade rows and from one to four radial flow splitters. Pure internal flow solutions were obtained as well as solutions with flow about the cowl/nacelle and various engine core flow conditions. The efficiency of the solution procedure was shown to be the same as the original analysis.

2. INTRODUCTION

This document contains the Computer Program User's Manual for the *ADPAC-APES* (Advanced Ducted Propfan Analysis Codes-Average Passage Engine Simulation) program developed by the Allison Gas Turbine Division of the General Motors Corporation under Task IV of NASA Contract NAS3-25270. The objective of this task is development of a three-dimensional flow analysis tool for advanced fan section and turbofan engine geometries such as the NASA/GE Energy Efficient Engine seen in Fig. 2.1 . The tool is able to compute steady flow solutions about geometries with any number of blade rows and any number of axisymmetric radial flow splitters. The tool simultaneously computes the flow through every flow path both internal and external, which includes the flow through the fan, and about the fan cowl and nacelle, from upstream to downstream of the engine. When the domain is extended in this manner, engine performance can be determined entirely by the analysis tool. Effects of engine core flow can also be simulated.

User instructions for setting up the computer program and running it for a demonstration geometry are found in Chapter three of this document. Chapter four covers some details on the subroutines, input and output of the program. Instructions and advice for running the code is given in Chapter five.

This flow analysis tool was developed from a code entitled *VSTAGE* which was

developed by John J. Adamczyk of the NASA Lewis Research Center [1] . The user is referred to the documentation for that code for additional information on the operation of the code.

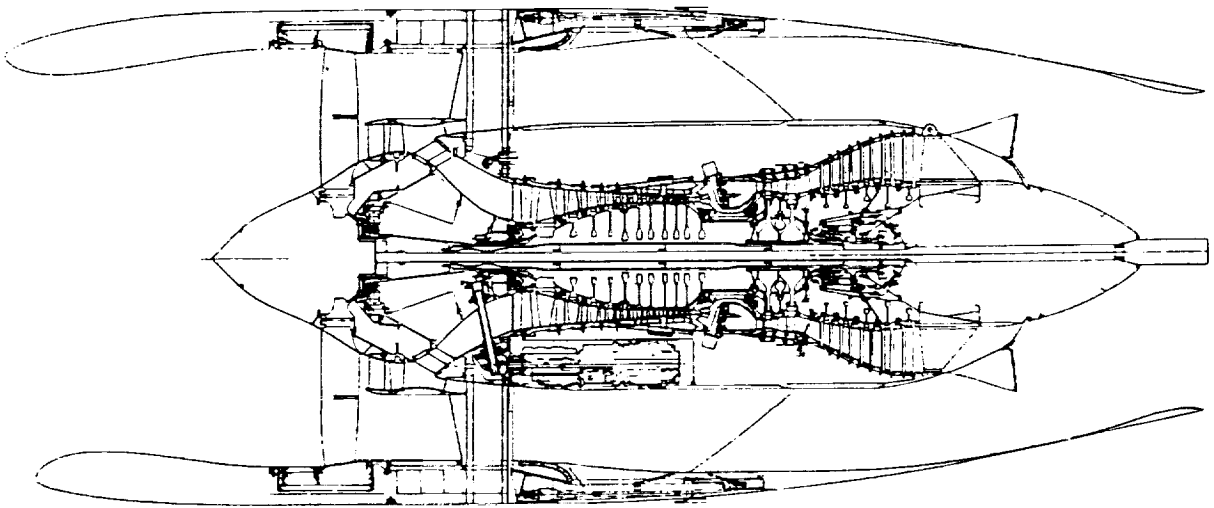


Figure 2.1: NASA/GE Energy Efficient Engine cross section

3. ADPAC-APES PROGRAM INITIALIZATION AND DEMONSTRATION

3.1 Introduction

This section describes the commands necessary to extract the *ADPAC-APES* source code from the standard distribution and run a complete test case for a flow solution about a ducted fan with a part span shroud. The standard *APES* distribution is normally a compressed *tar* file which can be decoded into the various parts by a sequence of commands on any standard UNIX system. The sequence listed below is intended to guide the user through the setup from the standard distribution and the running of the demonstration case.

3.2 Extracting the Source Files

The *APES* programs are distributed as a compressed *tar* file named

apes.tar.Z

The first step necessary to extract the *ADPAC-APES* programs is to uncompress the *tar* file with the command:

uncompress apes.tar.Z

This operation replaces the compressed file *apes.tar.Z* with an uncompressed file *apes.tar*.

The next step is to extract the individual files and directories from the *apes.tar* file. This process will create a subdirectory named *apes* in the current directory, so the user should move the *apes.tar* file to a suitable initial directory. Once the *tar* file is properly placed, the *ADPAC-APES* distribution may be extracted with the command

```
tar xvof apes.tar
```

(On some systems *tar xvf apes.tar* may be sufficient.) Execution of the command *ls -l* will verify that the *APES* directory has been created.

3.3 Compiling the Source Code

After extracting the source files, the user is naturally interested in compiling the source files for execution. A UNIX-compatible *Make* facility is provided for the *APES* program. The *Makefile* which governs the compilation process is necessarily machine-dependent and requires that the user select from one of a number of preconfigured systems. The systems which are immediately available are:

iris Silicon Graphics Iris workstation

cray Cray Computer Inc. supercomputer

aix IBM Aix operating system UNIX workstation

If no *system* is specified, then the *iris* system is assumed. The machine dependence

of the compilation process is inherently tied to the use of the Scientific DataBase LIBrary routines for binary file input/output.

In order to begin the compilation, it is first necessary to enter the *APES* directory with the command:

cd apes

At this point, several files and directories will be available. By entering the command **ls**, a listing of the individual directories can be obtained. The output of the **ls** command will look something like:

demo/ manual/ report/ sdblib/ src/

A description of each of these listings is given below:

demo This directory contains all the files necessary for generating a demonstration solution with the *ADPAC-APES* code.

manual This directory contains the *LaTeX* source code for this manual. If *LaTeX* is installed on your system, it is possible to reproduce this document (excluding figures) with the command **latex manual**. The resulting device independent file *manual.dvi* may then be converted to *PostScript* or previewed on screen through a number of widely available routines.

report This directory contains the *LaTeX* source code for the final report outlining the technical details of the *ADPAC-APES* codes. If *LaTeX* is installed on your system, it is possible to reproduce the final report (excluding figures) with the command **latex report**. Again the resulting file *report.dvi* may be converted to a print output or previewed.

src This directory contains all the FORTRAN source code for the *APES* program.

sdblib This directory contains the various machine-dependent files for the Scientific DataBase Library routines [2].

In order to compile the *APES* code, enter the *src* directory with the command:

```
cd src
```

The compilation is begun by issuing the command

```
make system
```

where *system* indicates the current computing platform (iris, cray, or aix) described above.

3.4 Running the Distribution Demonstration Test Case

Once the *make* facility has properly completed compiling the *APES* source code, it is possible to run the test case provided. It is recommended that the test case be run to verify proper compilation and extraction of the *APES* distribution.

In order to run the demonstration case, it is necessary to begin in the *demo* directory. At this point, the *demo* directory may be entered by issuing the command

```
cd ../demo
```

In the *demo* directory, an *ls* command will reveal:

```
demo.input1  grid1.demo  mover*
demo.input2  grid2.demo  script.flip*
```

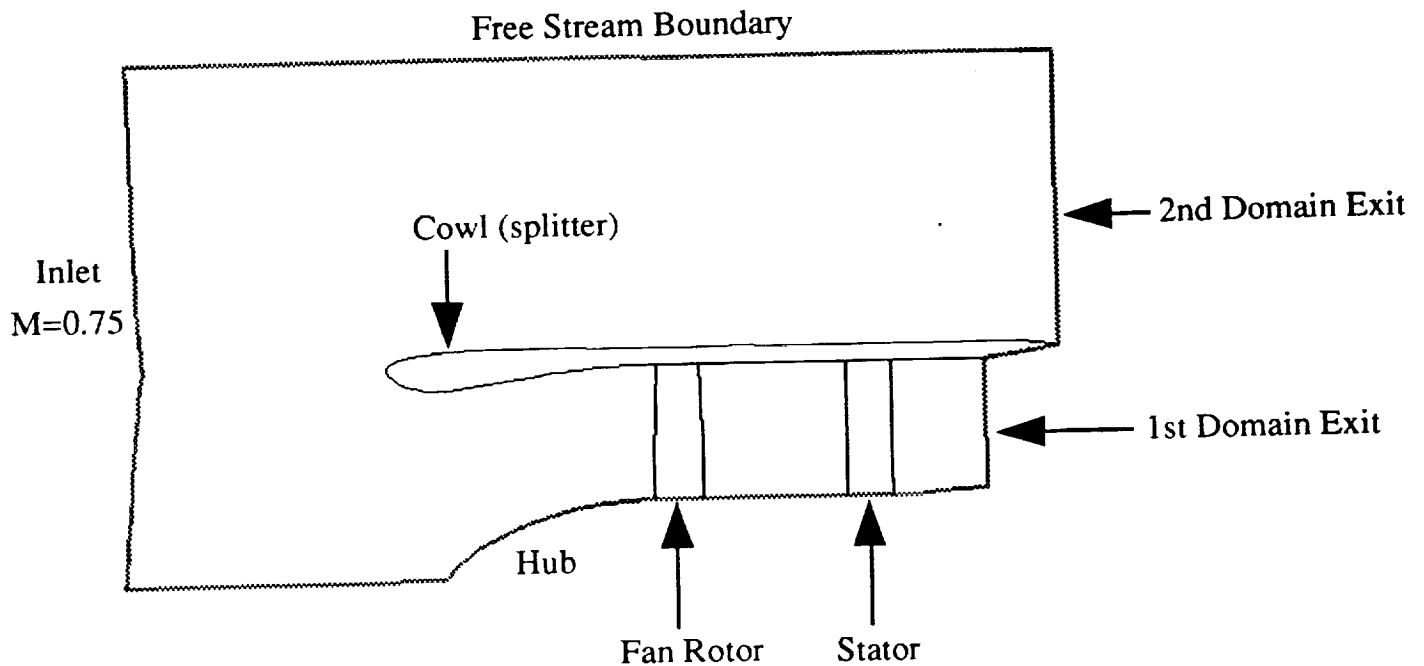



Figure 3.1: Demonstration case conditions and geometry

The demonstration case is a two blade row fan section with a cowl and a free stream outer boundary. Figure 3.1 shows the basic geometry and flow situation. The file *demo.input1* is the input file for the first blade row (rotor) of the demonstration case, and *demo.input2* is an input file for blade row number two (stator). The blade row specific grids for the rotor and the stator are files named *grid1.demo* and *grid2.demo*, respectively. The *mover** and *script.flip** files will be explained later. To start the demonstration case, the user can issue the command

```
../src/apes < demo.input1 > output1
```

This command executes the *APES* program, directing the input to be read from

the file *demo.input1* and directing the output to the file *output1*. When the program is finished, the *ls* command will reveal several new files:

```
axi1.new          bf1.new          flow1.new         flow1.new.hist
flow1.new.maxhist flow1.new.p3dabs  flow1.new.p3drel  flow1.new.rmshist
output1
```

The file *axi1.new* is the axisymmetric solution, the file *bf1.new* contains the blade body forces, and the file *flow1.new* contains the flowfield restart for the first blade row. All three files are needed to continue a multiple blade row calculation, while only *flow1.new* is needed for a single blade row calculation. A file containing the residual history data for the current run is *flow1.new.hist*. The residual history data files contain the maximum density time derivative (*flow1.new.maxhist*) and the root mean square average of the density derivative (*flow1.new.rmshist*) for each iteration. A *PostScript* plot of the history data can be generated using the *FULLPLOT* tool provided in the distribution (see the Appendix for a description of *FULLPLOT*). Before the plot file can be generated, the *FULLPLOT* code must be compiled by issuing the command:

```
f77 -o ../src/fullplot ../src/fullplot.f
```

When this has been done, a plot file of the residual history is produced by issuing the command:

```
../src/fullplot < flow1.new.hist
```

The resulting file *fort.15* can be previewed or plotted using a *PostScript* compatible printer. The files *flow1.new.maxhist* and *flow1.new.rmshist* contain the maximum

density residual and the average residual, respectively. The purpose of these files will be explained later. Finally, the files *flow1.new.p3dabs* and *flow1.new.p3drel* contain the absolute and relative *PLOT3D* flowfield data interpolated to the grid nodes, respectively. These files are formatted as *PLOT3D* multiple grid (mgrid=1) binary files and can be used in combination with the grid *grid1.demo* to examine the solution with *PLOT3D*.

To initialize and run the second blade row the user can issue the command:

```
../src/apes < demo.input2 > output2
```

The input file directs the code to initialize the stator blade solution with the axisymmetric solution of the rotor. The rotor blade's body forces (read in from the file *bf1.new*) are incorporated into the flowfield and the stator specific solution commences. When the program is finished, several more files will appear in the directory:

```
axi2.new          bf2.new          flow2.new          flow2.new.hist
flow2.new.maxhist flow2.new.p3dabs  flow2.new.rmshist  output2
```

These files contain the restart data, residual history data, and *PLOT3D* flowfield data for the stator blade row solution.

At this point one cycle through all the blade rows in this case (one "flip") has been completed. For this demonstration case, several flips are needed before a steady solution is obtained. See [3] or the **Solution Techniques** section of this manual for more information of flips. To set up for another flip, the user issues the command:

```
mover
```

This executes the file *mover** which contains several file manipulation commands which are simply executed in order. The commands rename the flowfield restart, axisymmetric solution, and body force files. Other commands append the blade row residual history files to a general residual history file so that a continuous record of the solution residuals can be plotted.

To execute a complete flip, the input files to *APES* must be changed so that the solution calculation can continue from its restart files. When this is done the commands:

```
../src/apes < demo.input1 >> output1
```

```
../src/apes < demo.input2 >> output2
```

mover

can run the demonstration case through one flip. All of these changes and commands have been included in the file *script.flip**. This file executes the program with the correct input files for each blade row and then rearranges all the necessary files to set up for the next flip. Thus, one flip can be accomplished by issuing the command:

script.flip

After several flips (about 10) the solution should be converged. The solution is judged as converged when the root mean square average of the density residual (i.e., the quantity in the file with the suffix *new.rmshist*) has been reduced by a factor of 10^{-3} from its initial value. For more information about flips and convergence can be found in the Solution Techniques Section.

4. ADPAC-APES PROGRAM GENERAL DESCRIPTION

4.1 Parameter/Array Sizes

It is necessary to estimate the maximum array sizes and other limit parameters prior to the compilation process. Most of the arrays in the routines of the code are dimensioned according to the PARAMETER statements in the file *parameter.inc* which contains the line:

```
PARAMETER ( IMX=100, JMX=50, KMX=50, MSP=6 )
```

These parameters dimension numerous arrays found in the subroutines. The PARAMETER variable are defined as.

IMX Number of streamwise grid lines + 1

JMX Number of spanwise grid lines + 1

KMX Number of circumferential grid lines + 1

MSP Maximum number of spanwise flow splitters

Before running the code, the user should know the size of the grid and make sure the array sizes are adequate. The PARAMETER variables can be larger than definitions above but errors and peculiar results will occur if the arrays are dimensioned below the required parameter.

4.2 Input File Description - Basic Input Section

A sample input data file for *ADPAC-APES* is shown in Fig. 4.1. The beginning of the input file consists of an arbitrary number of header lines (for user information) which is terminated by a line beginning with a "+" character. Following this, title information is read in and will be printed out in the output file. There can be up to nine lines of title information. If more than nine lines are present, then only the first eight lines of information and the last line will be output from the code. As with the header section, the title information is terminated by a line beginning with a "+" character. Beyond this line, the file follows the structured format which is seen in Fig. 4.1, beginning with the "+-MACH" characters. The data is structured as to the order it is read in but there is no field oriented format in which the input must be organized. The field markers seen in Fig. 4.1 are only to assist the user as to the order of the input data. Most input data is read in as a real number except where indicated. A brief description of the variables used in the data file is given below. Some of the variables in the input file are not used in the current code but their place is kept so that continuity with other flow solvers is maintained. Values are needed for these inactive variables since they hold places and are read in before active variables. The inactive variables are italicized in the descriptions below.

VARIABLE DESCRIPTION

MACH	Freestream Mach number.
GAMMA	Specific heat ratio (c_p/c_v).
PEXIT	Exit static pressure. This is applied at the hub and

```

User information about this particular blade row or running condition

+----- TITLE ----( up to 9 lines )-----+
DEMONSTRATION INPUT FILE  FOR A GENERIC FAN ROTOR
THIS FAN ROTOR IS ONE BLADE ROW IN A THREE BLADE ROW CALCULATION
THERE IS ONE SPLITTER (CORE/BYPASS) AND A CORE CONDITION
REPRESENTING THE TURBINE EXIT IS SIMULATED
THE GRID IS 150 BY 50 BY 50 FOR THIS BLADE ROW
TAKE OFF CONDITIONS ARE MODELED FOR THE SYSEM

+---MACH---+---GAMMA---+---PEXIT---+---OMEGA---+---ADVR---+---DUCT---+---BCFAC---+
0.3500    1.4000    0.0000   -4500.000    0.00    1.0    1.0000
+---CFL---+---VIS2---+---VIS4---+---QFIL---+---BC---+---BLDROW---+---DTAXI---+
-4.0000    2.00    1.000    0.0    0.0000    1.0    1.00
+---FNCMAX---+---REST---+---SAVE---+---FISTEP---+---FNPRNT---+---ROWMAX---+---FPRAXI---+
100.0    -1.0    -1.0    1.0    0.0    3.0    00.0
+---REFINI---+---REFINJ---+---REFINK---+---EPSX---+---EPSY---+---EPSZ---+---FPRBF---+
1.0    1.0    1.0    2.00    2.00    2.00    0.0
+---RFINII---+---RFINJ---+---RFINKK---+---FAXI---+---FAXIMX---+---FR3DMX---+---FTURB---+
1.0    1.0    1.0    0.0    99999.0    0.0    0020.0
+---FSTRT3---+---FEXIT3---+---FSET---+---FINVIS---+---JBASE---+---JTIP---+---fwfcj---+---fwfck---+
0.0    0.0    0.00    0.0    1.0    53.0    1.0    1.0
+---DIAM---+---TREF---+---PREF---+---RGAS---+---PR---+---PRT---+---trans---+---itrans---+
5.5750    518.7    2116.8    1716.48    0.72    0.90    0.0    1.0
+---OMEGS1---+---FCHL---+---FCHT---+---OMEGS2---+---FCTL---+---FCTT---+---fiwfs---+---fiwfe---+
-4500.00    9.0    150.0    0.0    0.0    0.0    10.0    150.0
flow.rest.for1    blade_force.for1    axi_flow.for1    grid.for1
flow.rest.for2    blade_force.for2    axi_flow.for2    grid.for2
flow.rest.for3    blade_force.for3    axi_flow.for3    grid.for3
flow.rest.new1    blade_force.new1    axi_flow.new1
+---NEXITS---+---LCORE---+---NMAVG---+---LSPLIT---+---LINLET---+
1    1    2    1    1
+---PEXIT1---+---PEXIT2---+---PEXIT3---+ ( EXIT PRESSURES )
1.310
+---ICORIN---+---ICOREX---+---JCORE---+---FCORE---+ ( ENGINE CORE SIMULATION VARS )
141    141    19    1.0
+---PSCORE---+---PTCORE---+---TTCORE---+---ANCORE---+ ( ANGLE IN DEGREES )
1.15    1.0    1.0    0.0
+---IAVG---+---J1AV---+---J2AV---+ ( MASS AVG. PLANE STREAMWISE & SPANWISE BOUNDS )
123    1    19
137    21    59
+---NSPLIT---+ ( NUMBER OF RADIAL SPLITTERS )
1
+---JSPL---+---ISPLE---+---ISPTE---+ ( SPLITTER SPANWISE & STREAMWISE BOUNDS )
19    75    141
+---LPITFLG---+---LTTFLG---+---LANFLG---+---NINPTS---+ ( INLET PROFILE PARAMETERS )
0    0    0    -3
+---PTIN---+---TTIN---+---ANIN---+---RPOS---+
2116.0    520.0    10.0    0.10
2326.0    420.0    20.0    0.50
2116.0    320.0    30.0    0.90

```

Figure 4.1: Sample input data file for ADPAC-APES

integrated radially outward for shrouded geometries
(DUCT=1.0), or applied at the outer exit boundary and
integrated radially inward for propellers (DUCT=0.0),
to satisfy radial equilibrium.

(this parameter may not be used when domain has multiple
exits, see exit pressure definition in Extra Input Section)

OMEGA Rotational speed (revolutions per minute) of the
particular blade row. Positive value indicates rotation
counter-clockwise as one is looking down the machine axis.

ADVR Not active in this version.

DUCT Internal flow duct option:
if = 0.0, external flow options are utilized (unshrouded blade with free
stream outer boundary),
if = 1.0, an internal flow is assumed (compressor).
Solid wall boundary conditions are applied at the outer
boundary rather than the characteristic far field
condition.

BCFAC Not active in this version.

CFL Time step parameter:
if ≤ 0 then this is the CFL number used
to determine the calculation time step for
local time stepping (i.e., for each cell),
if > 0.0 then this is the CFL number used

to determine the calculation time step without local time stepping (i.e., most restrictive cell CFL sets time step throughout domain).

VIS2 Second order damping coefficient ($\approx 1.0 - 3.0$)

(divided by 4 in the code)

VIS4 Fourth order damping coefficient ($\approx 1.0 - 2.0$)

(divided by 64 in the code)

QFIL Artificial dissipation trigger:

if ≤ 0.0 then the standard dissipation scheme is used.

if ≥ 1.0 then the dissipation is directionally scaled.

(this eigenvalue scaling is sometimes usefull for highly stretched grids)

BC Exit boundary condition trigger:

if $= 0.0$, characteristic boundary condition based on P_{tot} is used,

if $= 1.0$, characteristic boundary condition based on mass flow is used.

BLDROW Blade row parameter. This value determines which blade row

is being calculated during a multiple blade row solution.

For single blade row calculations, this should be $= 1.0$.

DTAXI Not active in this version.

FNCMAX Maximum number of time steps to be performed.

REST Restart option parameter:

if $= -1.0$ a restart file and body force file are read,

(multiple blade row configuration)

if $= 0.0$ no restart file, initialize variables in code,

if = 1.0 a restart file is read.

(single blade row configuration)

SAVE Save restart file option parameter:

if = -1.0 restart files and body source terms are output

at the end of a run (multiple blade row configuration),

if = 0.0 no restart file is output at the end of the run,

if = 1.0 a restart file is output at the end of the run.

(single blade row configuration)

FISTEP Number of iterations performed between recalculation of the time step.

FNPRNT Flowfield output parameter:

if = 0.0 no flowfield data printed out,

if = 1.0 inlet flow (to leading edge) quantities written to output file.

if = 2.0 flowfield written out in PLOT3D multiple

grid, binary Q file format.

ROWMAX Maximum number of blade rows in the current configuration.

For single rotation, = 1.0. For multiple blade rows,

this parameter indicates that additional grids and

body source terms are to be read.

FPRAXI Print axisymmetric solution,

if = 0.0 axisymmetric flow quantities are not printed,

if = 1.0 axisymmetric flow quantities (ASCII) are written to the output

file.

REFINI Not active in this version.

REFINJ Not active in this version.

REFINK Not active in this version.

EPSX Implicit residual smoothing coefficient in the streamwise direction (≈ 2.0 is a typical value).

EPSY Implicit residual smoothing coefficient in the spanwise direction (≈ 2.0 is a typical value).

EPSZ Implicit residual smoothing coefficient in the circumferential direction (≈ 2.0 is a typical value).

FPRBF Print trigger for body forces:
 if = 0.0, body forces are not printed,
 if = 1.0, body forces (ASCII) are written to the output file.

RFINII Not active in this version.

RFINJJ Not active in this version.

RFINKK Not active in this version.

FAXI Number of axisymmetric solutions, not active in this version (0.0).

FAXIMX Number of 3D passes before axisymmetric solutions, not active in this version (99999.0).

FR3DMX Number of reduced domain 3D passes, not active in this version (0.0).

FTURB Number of iterations performed before recalculating the turbulent viscosity.

FSTRT3 Not active in this version (0.0).

FEXIT3 Not active in this version (0.0).

FSET Flowfield restart parameter:

if = 0.0, standard restart procedure used,
 if = 1.0, restart using the previous axisymmetric solution,
 if = 2.0, restart using a piecewise construction of the
 flowfield from the axisymmetric solution of the other blade rows,
 if = 3.0, restart with a redefined uniform flowfield from
 inlet to the blade row leading edge and from the trailing
 edge to the exit.

FINVIS Viscous/inviscid solution trigger. This variable determines whether a viscous or inviscid solution is performed:

if = 1.0, inviscid;

if = 0.0, viscous.

JBASE User defined base of the blade row.

JTIP User defined tip of the blade row. This overrides the spanwise tip index found in the grid file.

NOTE: The parameters **JBASE** and **JTIP** specify the bounds from which flow passes over and under the blade (a periodic surface), regardless of whether or not the mesh defines a periodic surface at those bounds.

FWFCJ Wall function trigger for the endwalls.

if = 0.0, shear stress near endwalls found from velocity gradient,

if = 1.0, wall function used to determine shear stress near the endwalls and splitters.

FWFCK Wall function trigger for blade surfaces.

if = 0.0, shear stress near blades found from velocity gradient,
if = 1.0, wall function used to determine shear stress
near the blade surfaces.

DIAM The rotor diameter or the value of the nondimensionalizing
factor used in the input grid (feet).

TREF Reference total temperature (degrees Rankine).

PREF Reference total pressure (pounds per square foot).

RGAS Gas constant (foot-pounds per slug degree Rankine).

PR Gas Prandtl number.

PRT Turbulent Prandtl number (0.9 recommended).

TRANS Fraction of the chord from the leading edge that transition
on the suction surface begins.

NOTE: All other surfaces have turbulent boundary layers.

ITRANS The suction surface indicator for the transition model:

if = 1.0, the suction surface is the first circumferential index of the
grid (usually if **OMEGA** is negative),

if = 2.0, the suction surface is the last circumferential index of the grid
(usually if **OMEGA** is positive).

OMEGS1 Rotational speed of a user specified section of the hub (RPM).

FCHL Grid index for the hub or spinner leading edge (see Fig. 4.2) rotating
at OMEGS1 (used only in viscous calculations).

FCHT Grid index for the hub trailing edge (see Fig. 4.2) rotating at OMEGS1
(used only in viscous calculations).

OMEGS2 Rotational speed of a second user specified section of the hub (RPM).

FCTL Grid index for the hub leading edge (see Fig. 4.2) rotating at OMEGS2
(used only in viscous calculations).

FCTT Grid index for the hub trailing edge (see Fig. 4.2) rotating at OMEGS2
(used only in viscous calculations).

FIWFS Axial grid index at which the wall function model starts.

FIWFE Axial grid index at which the wall function model ends.

4.3 Input File Description - Blade Row File Name Section

The section following the basic input parameters contains the names of files needed to restart a solution and the names of files written out following the solution iteration procedure. That section of the input file with suggested file names is reproduced below. For each blade row in the domain (= BLDROW parameter) a line in the input file is required. This line contains four names of files pertinent to that blade row solution. The first file name is the flowfield restart for that blade row. The second name is for the blade row body force file. The third name is for the axisymmetric flowfield, and the fourth is for the grid particular to that blade row.

<code>flow.rest.for1</code>	<code>blade_force.for1</code>	<code>axi_flow.for1</code>	<code>grid.for1</code>
<code>flow.rest.for2</code>	<code>blade_force.for2</code>	<code>axi_flow.for2</code>	<code>grid.for2</code>
<code>flow.rest.for3</code>	<code>blade_force.for3</code>	<code>axi_flow.for3</code>	<code>grid.for3</code>
<code>flow.rest.new1</code>	<code>blade_force.new1</code>	<code>axi_flow.new1</code>	

Following the line(s) pertaining to blade row restart data, one line is required for the outputted restart data. Names for the updated restart flowfield, blade forces,

No current shroud rotation capability

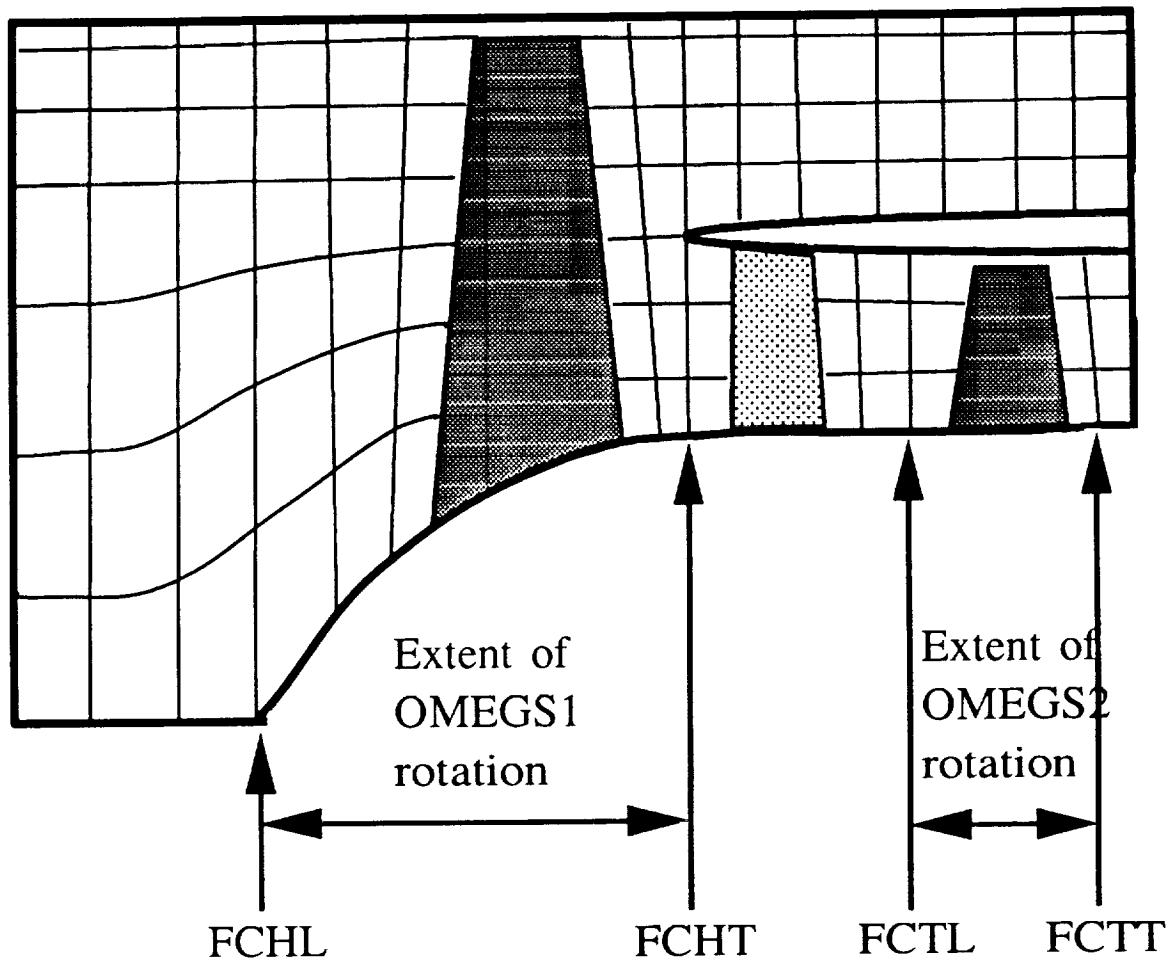


Figure 4.2: Fan section displaying hub rotation regions/parameters

and axisymmetric solution for the current blade row are on this line. The names can be up to thirty characters long but all four (or three) names must not take up more than one eighty column line. The names should be separated by blanks or commas.

4.4 Input File Description - Extra Input Parameter Section

The third and last section of the input file contains parameters important to multiple splitter calculations, engine simulations, and some user specified post processing. The parameters are read in a free format and are both integer and real. Explanations of the parameters follow and a number of these are illustrated graphically in Fig. 4.3.

VARIABLE DESCRIPTION

NEXITS Number of flowfield domain exits (int). If one or more flow splitters extends to the exit, the domain exit is broken up (Fig. 4.3) and boundary conditions must be independently specified for each exit section. NEXITS specifies how many non-core (see LCORE below) exit sections there are.

LCORE Core boundary condition flag (int).

If = 0, no core is simulated.

If = 1, a core inlet (and possibly exit) is simulated. A core inlet is treated as a flowfield domain exit with appropriate boundary conditions. A core inlet is sometimes defined at the last axial grid index (see Fig. 4.3) instead of a standard exit because the output file includes core mass flow in its iteration history when a core is specified.

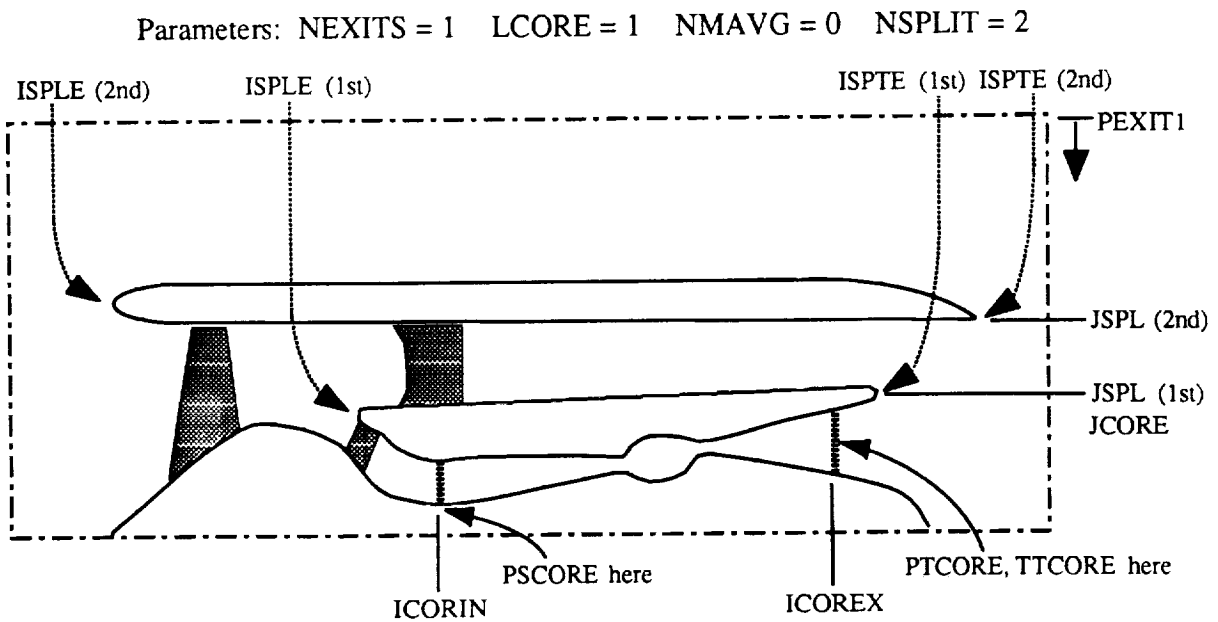
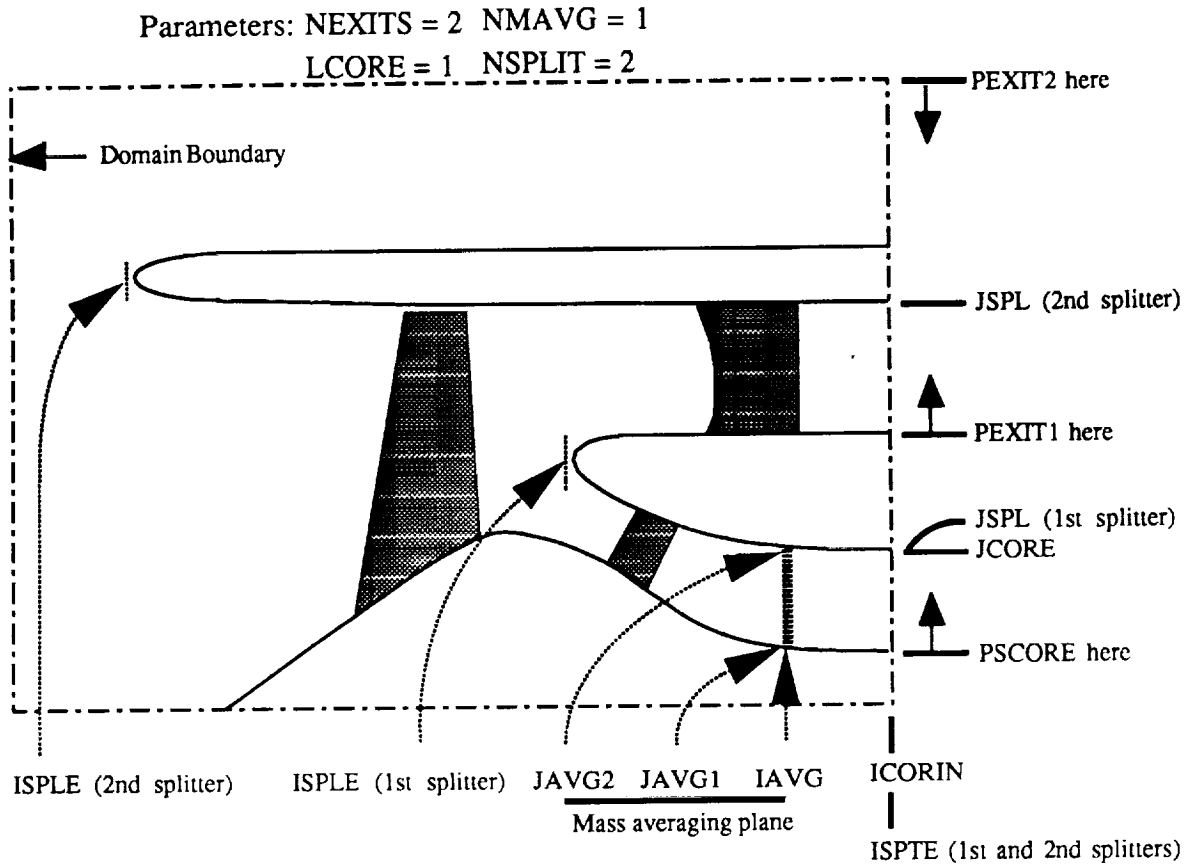


Figure 4.3: Computational domain highlighting some input parameters

NMAVG The number of mass averaged streamwise computational planes (int).
This is a post-processing parameter to specify the number of streamwise grid planes (I index=const) for circumferential mass averaged and global mass averaged data.

LSPLIT Flow splitter definition flag (int).
If = 0, no splitters are defined by the user.
If = 1, splitter bounds are input by the user.

LINLET Inlet condition definition flag (int).
If = 0, standard inlet conditions are used.
If = 1, user defines inlet conditions/profile.

PEXIT1 Exit static pressure for the first exit section (real). This is applied at the lower endwall and integrated radially outward, or applied at the outer exit boundary and integrated radially inward (if the exit section has a free stream outer bound), to satisfy radial equilibrium. See PSCORE definition.

PEXIT2 Exit static pressure for the second exit section etc.

ICORIN Streamwise grid index for the core inlet (int).

ICOREX Streamwise grid index for the core exit (int).

JCORE Spanwise grid index for the top of the core passage (int). The bottom of the core passage is the first spanwise index.

FCORE Not currently used (real).

PSCORE Core inlet static pressure at the hub (real).

Note: If a core is specified then PSCORE sets that domain exit pressure

and the next (spanwise) domain exit pressure will be set by PEXIT or PEXIT1.

PTCORE Core exit total pressure (real).

This parameter has no effect on the flowfield if ICOREX is greater than the last streamwise grid index.

TTCORE Core exit total temperature (real).

This parameter has no effect on the flowfield if ICOREX is greater than the last streamwise grid index.

ANCORE Core exit flow angle in degrees from axial (real).

This parameter has no effect on the flowfield if ICOREX is greater than the last streamwise grid index.

IAVG Streamwise grid index for a mass average data plane (int).

J1AV Spanwise grid index where the mass average plane begins (int).

J2AV Spanwise grid index where the mass average plane ends (int).

NSPLIT Number of flow splitters (int).

JSPL Spanwise grid index at the lower surface of the splitter (int).

ISPLE Streamwise grid index at the leading edge of the splitter (int).

ISPTE Streamwise grid index at the trailing edge of the splitter (int).

LPNFLG Inlet total pressure trigger (int):

If = 0, a constant total pressure (reference pressure) profile is used.

If = 1, a spanwise total pressure profile is input (lbs/square foot).

If = 2, a spanwise total pressure profile is input (nondimensional Pt).

LTNFLG Inlet total temperature trigger (int):

If = 0, a constant total temperature (reference temp) profile is used.

If = 1, a spanwise total temperature profile is input (degrees Rankine).

If = 2, a spanwise total temperature profile is input (nondimensional Tt).

LANFLG Inlet absolute swirl angle trigger (int):

If = 0, a constant swirl angle (0 deg from axial) profile is used.

If = 1, a spanwise swirl angle profile is input (deg from axial).

If = 2, a spanwise swirl angle profile is input (angle in radians).

NINPTS Number of points in the spanwise profile (int).

If > 0, dimensional spanwise positions used (feet).

If < 0, fractional spanwise positions used (hub= 0.0 to shroud = 1.0).

PTIN Inlet total pressure (real).

TTIN Inlet total temperature (real).

ANIN Inlet absolute flow angle (real).

RPOS Spanwise position of inlet condition data (real).

4.5 Output File Description

ADPAC-APES produces one or more output files depending on the parameters set in the input file. The user is referred to the previous sections for definitions of all the possible output that can be written. Of particular importance to restarting a solution or running a multiple blade row solution are the files listed in the Input File Description-Blade Row File Name Section. A flowfield restart, blade body force, and axisymmetric flowfield are written out using the Scientific DataBase Library routines [2]. These files are in a machine independent format, and are compatible with the PLOT3D multiple-grid, binary file description.

Much of the output is written to the standard output device, which typically would be the system monitor, or can be redirected to a file in a unix system with the statement structure shown below.

apes < input.file > output.file

This statement will send the *ADPAC-APES* output to the file *output.file*. This output consists of a print out of the input parameters, messages of actions taken during the run (e.g., writing out a restart file), and a print out of the residuals and certain flowfield information at every iteration. An example of the "iteration history" portion of the output file is seen in Fig. 4.4. At the end of each iteration or step, some residuals of the flowfield variables are statistically manipulated and printed out. At each step the maximum value of the density residual (which can be thought of as a time derivative of the change in density) is printed out. The location of this maximum is also printed out and is sometimes useful in determining a region of instability in

```

SOLUTION RESTARTED FOR BLADE ROW 1 AT N = 300      TIME = 1279.3789
-----
eddy:  cpu =2.106373572 seconds
step  max(dr/dt)  i   j   k  rms(dr/dt)  rssq(q)  mf(in)  mf(out)  mf(core)
-----
301  0.130E+02  47  54  2  0.251E+00  0.279E+00  289.73  287.30  38.32
302  0.126E+02  47  54  2  0.252E+00  0.280E+00  289.74  287.24  38.32
303  0.123E+02  47  54  2  0.251E+00  0.279E+00  289.74  287.18  38.31
304  0.119E+02  47  54  2  0.251E+00  0.278E+00  289.74  287.11  38.31
305  0.115E+02  47  54  2  0.251E+00  0.279E+00  289.74  287.05  38.30
306  0.110E+02  47  54  2  0.251E+00  0.279E+00  289.74  286.98  38.30
307  0.105E+02  47  54  2  0.251E+00  0.279E+00  289.74  286.92  38.29
308  0.100E+02  47  54  2  0.252E+00  0.279E+00  289.74  286.85  38.29
309  0.945E+01  47  54  2  0.253E+00  0.280E+00  289.74  286.78  38.29
310  0.891E+01  47  54  2  0.254E+00  0.280E+00  289.74  286.72  38.28
.      .      .      .      .      .      .      .      .
.      .      .      .      .      .      .      .      .
.      .      .      .      .      .      .      .      .

```

Figure 4.4: Portion of output file showing iteration history

the flowfield or a poorly gridded region. The root mean square of the density residual is printed next. Following this the square of the change in all the state variables is summed and the root of this is printed ($\text{rssq}(q)$). The values printed after the residuals depend upon the geometry of the solution. If the geometry is unshrouded and there are no flow splitters, then a power coefficient, a thrust coefficient, and the number of supersonic flowfield points are printed. For shrouded geometries with no core defined, the inlet mass flow, exit mass flow, and the number of supersonic flowfield points are printed. For shrouded geometries with a core, the inlet, exit, and core corrected mass flow are printed (see Fig. 4.4).

Three other output files are produced and they also have information on the residual history in them. They are plot files and a *PostScript* image can be generated

with the *FULLPLOT* program developed under the same NASA contract [4]. A description of *FULLPLOT* is reprinted from [4] in the Appendix. One file contains the history of the maximum density residual, another contains the history of the root mean square of the density residual, and the third contains both values. The program assigns a name to these files by adding a ".hist" or ".maxhist" or ".rmshist" suffix to the flowfield restart file name. For example, if the flowfield restart has the name

flow.rest

then the iteration history plot file of the density residual's root mean square will be named

flow.rest.rmshist

Some of the output that the user can specify (through input parameters described in an earlier section) is printed in the nondimensional form used by the flow solver directly, and some output is printed out in dimensional form using the user set reference values. So that the user can understand and manipulate the output data better, a description of the flow variable nondimensionalization is presented.

Some of the variables in the numerical solution are nondimensionalized by reference values as follows:

$$z = \frac{\bar{z}}{L_{ref}}, \quad r = \frac{\bar{r}}{L_{ref}}, \quad v_z = \frac{\bar{v}_z}{v_{ref}}, \quad v_r = \frac{\bar{v}_r}{v_{ref}}, \quad v_\theta = \frac{\bar{v}_\theta}{v_{ref}}$$

$$p = \frac{\bar{p}}{p_{ref}}, \quad T = \frac{\bar{T}}{T_{ref}}, \quad \rho = \frac{\bar{\rho}}{\rho_{ref}},$$

The quantities are defined as follows:

- p The nondimensional pressure.
- r The nondimensional radial length.
- T The nondimensional temperature.
- v_r The nondimensional radial velocity.
- v_z The nondimensional axial velocity.
- v_θ The nondimensional tangential velocity.
- z The nondimensional axial length.
- ρ The nondimensional density.
-
- L_{ref} The reference length nondimensionalizing the grid.
This should be the input parameter **DIAM**.
- p_{ref} The reference (or freestream) relative total pressure
This should be the input parameter **PREF**.
- T_{ref} The reference (or freestream) temperature
This should be the input parameter **TREF**.
- v_{ref} The reference (or freestream) velocity determined from the
relative total conditions $v_{ref} = \sqrt{p_{ref}/\rho_{ref}}$
- R_{ref} The reference (or freestream) gas constant
This should be the input parameter **RGAS**.
- ρ_{ref} The reference (or freestream) relative total density
determined from the relative total conditions

$$\rho_{ref} = \frac{p_{ref}}{T_{ref} R_{ref} / g}$$

4.6 Subroutine Description

A list of the *ADPAC-APES* subroutines and their functions is given below for reference. A skeleton program flowchart is illustrated in Fig. 4.5.

SUBROUTINE DESCRIPTION

APES	Main calling routine.
BAXI	Calculates axisymmetric flow quantities.
BCCORE	Boundary condition for engine core simulation.
BCEXT	Boundary condition routine for exit cells.
BCINL	Boundary condition routine for inlet cells.
BCJV	Boundary condition routine for j =constant surfaces (hub and outer boundaries).
BCK	Inviscid boundary condition routine for the blade surfaces.
BCKV	Viscous boundary condition routine for the blade surfaces.
BCSPL	Boundary condition routine for the hub, shroud, and flow splitter surfaces.
BCSPL2	Axisymmetric boundary condition routine for the hub, shroud, and flow splitter surfaces.
BCWAL2	Axisymmetric boundary conditions at the exit for the 2D solution.
BCWALL	Boundary condition routine for periodic cells.
BFORC1	Routine for calculating the axisymmetric average of the flowfield residuals.

BFORC2 Routine for calculating body force terms using the axisymmetric equations.

BFORC3 Routine for calculating viscous terms needed in BFORC1.

BFORC4 Routine for calculating viscous terms needed in BFORC2.

BFREAD Routine to read in body force terms for all adjacent blade rows.

CONATAN Continuous tangent function evaluation routine.

CONVAS Array storage conversion routine used in conjunction with SDBLIB machine-independent input/output routines.

EDDYDW Turbulence model evaluation routine.

EDYDW2 Axisymmetric turbulence model evaluation routine.

ERROR Convergence-checking routine for Runge-Kutta solver.

EULER Runge-Kutta solver.

FILTER Artificial dissipation routine.

FIXUP Routine to correct corner phantom cells so output averages and viscous stress evaluations are correct.

FLREAD Routine to read in the flowfield restart file.

FSTR2 Axisymmetric viscous stress evaluation along i coordinate direction.

FSTRES Viscous stress evaluation along i coordinate direction.

GETMAS Routine to compute mass average quantities at an $i=\text{constant}$ coordinate surface.

GRIDG Routine to read and set up the grid.

GSTR2 Axisymmetric viscous stress evaluation along j coordinate direction.

GSTRES Viscous stress evaluation along j coordinate direction.

HSTR2 Axisymmetric viscous stress evaluation along k coordinate direction.

HSTRES Viscous stress evaluation along k coordinate direction.

INIT Routine to initialize flowfield with reference flow quantities.

INITA Routine to initialize flowfield with axisymmetric solution.

INITB Routine to replace flowfield with a piecewise constructed one from the latest axisymmetric solutions.

INITC Routine to redefine flowfield from inlet to blade leading edge and from trailing edge to exit.

INITD Routine to reset inlet conditions based on the previous blade row axisymmetric solution.

INPUT Routine to read in input data and set up reference values.

METRI2 Routine to calculate cell volumes and surface normals for 2D solution.

METRIC Routine to calculate cell volumes and surface normals.

OUTPUT Routine to print output and save restart files.

P3DOUT Routine to output *PLOT3D* flowfield (Q) files.

PLHIST Routine to output convergence history plot files.

PTCALC Routine to determine inlet boundary conditions.

REFIN Routine to refine a 3D solution.

REFIN2 Routine to refine a 2D solution.

REFIN2B Routine to interpolate blade body forces to a finer grid.

REFIN3 Routine to interpolate flowfield to a finer grid.

REFINB Routine to interpolate beta to a finer grid.
RESID Implicit residual smoothing routine.
RUNGE Runge-Kutta flux calculation routine.
SAREA Routine to determine metric area terms.
SAREA2 Routine to determine axisymmetric metric area terms.
STEP Routine to determine time step.
TRIM Routine to delete trailing blanks from a character string.
STEP2 Routine to determine time step for 2D solution.
VOLUX Routine to calculate an individual cell volume.
VOLUX2 Routine to calculate an individual 2D cell volume.
WF Wall function turbulent stress routine.

It should also be mentioned that a number of routines from the SDBLIB library are also included in the source code distribution, but are not defined here.

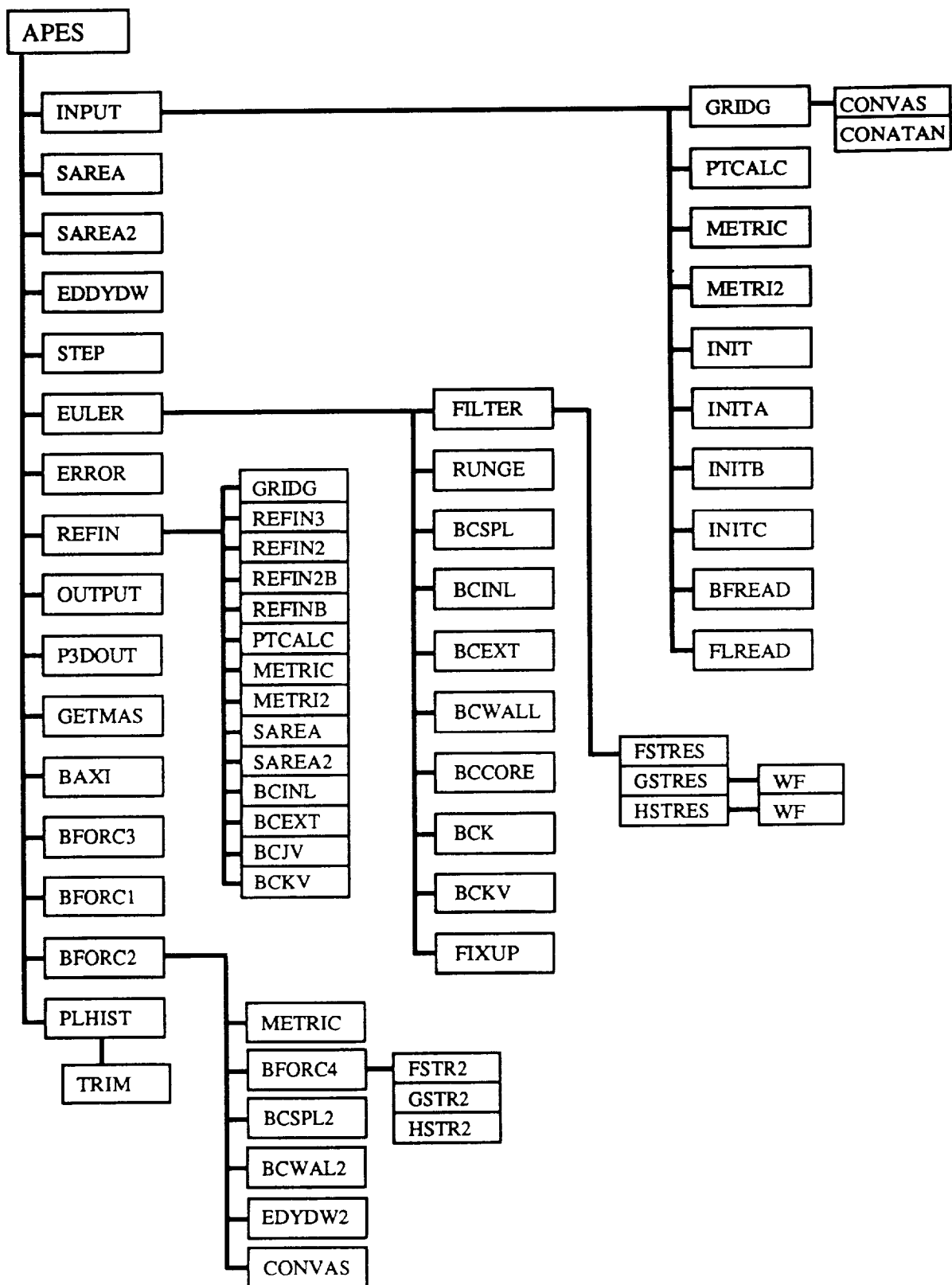


Figure 4.5: Subroutine flowchart of ADPAC-APES

5. ADPAC-APES SOLUTION TECHNIQUES

5.1 Single/Multiple Blade Row Procedure

A procedure for obtaining a numerical solution for multiple blade rows with the *APES* program is described below. The single blade row case is in general a reduction of the multiple blade row case and is described later.

Before executing the solution algorithm, numerical grids (one for each blade row) are required. These grids model the actual three-dimensional geometry of their particular blade row, and represent the rest of the domain as an axisymmetric duct. *APES* requires all grids to have the same meridional representation (i.e., the same dimensional (z,r) coordinate lattice structure). If the geometry contains radial flow splitters (e.g., a cowl), the spanwise grid surface index is duplicated for the points defining the lower split surface. This is done so that the phantom cells defined in the flow solver will have a grid index associated with them. Figure 5.1 shows this spanwise index relationship. Grids produced by the *TIGG3D* program fulfill these requirements and more information on the grids and *TIGG3D* can be found in [5].

The solution procedure is begun from a set of initial data. This initial data is specified as a uniform flow (based upon the freestream Mach number), or may be introduced from a previous solution. The type of initialization selected, as well as all

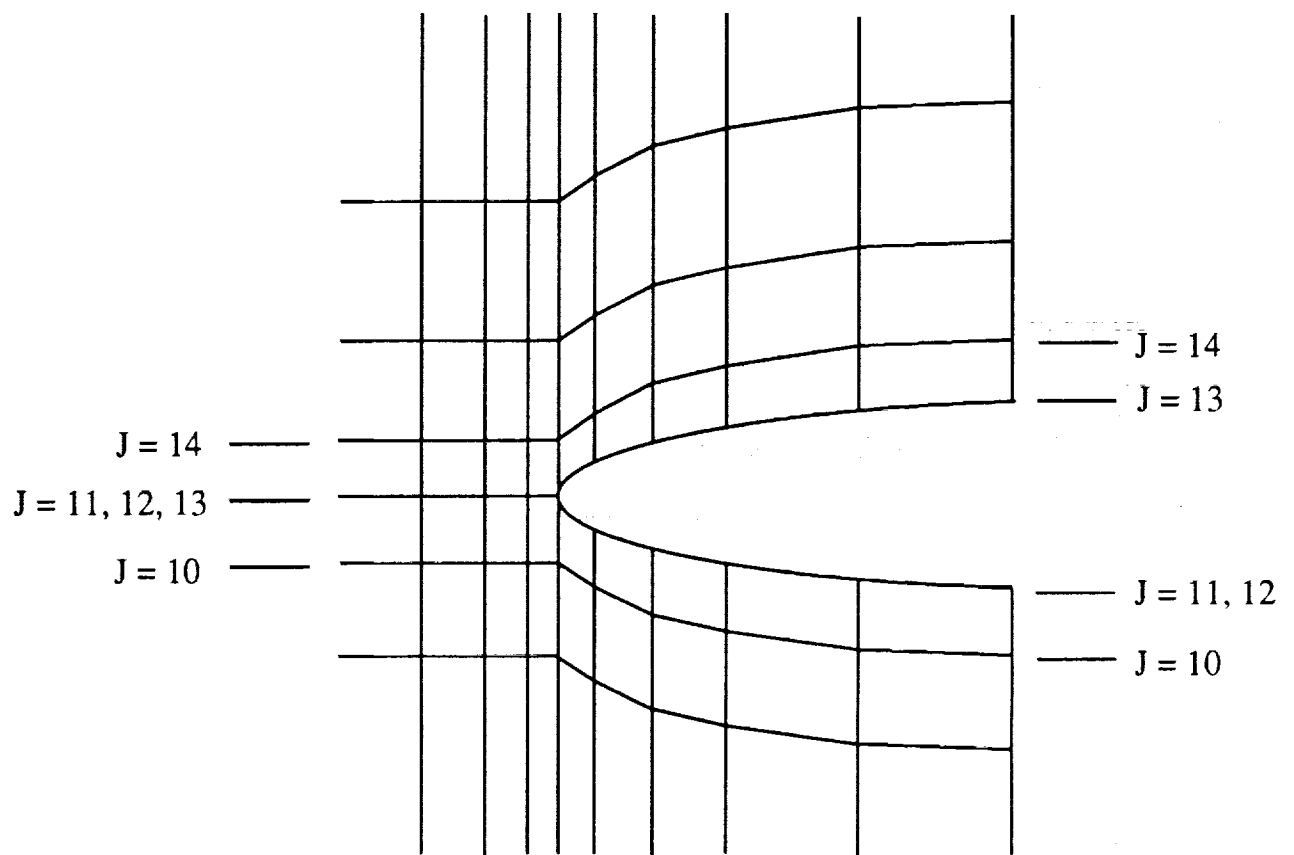


Figure 5.1: Spanwise index numbering around radial splitters

other program options, is controlled by parameters in the input file (see the Input File Description sections). For uniform flow initialization, the parameter **REST** must be zero and the parameter **FSET** should be zero as well. If a previous solution is used, the parameter **REST** is set to one. This procedure is typical for either a single blade row solution or the first blade row of a multiple row solution. For initialization of the blade row solutions following the first, the parameter **REST** is set to zero in those blade row input files. This will cause the initial flowfield to be computed from the axisymmetric solution of the previous blade row. Also, blade body forces from the previously run solutions will be incorporated in the current blade row calculation. The axisymmetric solution and body forces are read from file names listed in the input file. A diagram of this procedure for a three blade row solution is seen in Fig. 5.2.

Once the initialization has occurred, a multiple blade row solution is found using a nested iteration procedure with an inner and outer loop as seen in Fig. 5.3. The inner loop represents the iterative technique used to solve the flow equations for a particular blade row (a single *APES* run). This *APES* run for a individual blade row incorporates the effects of the neighboring blade rows (input through the body force file names). The run iterates a number of steps (marches in time to a steady solution) based on the parameter **FNCMAX**. For the *APES* restart/run the parameter **REST** must be set to negative one. When each blade row has gone through this iteration and the blade row effects have been recalculated, one cycle or "flip" through the system (outer loop) is complete. When these updated effects (which are calculated automatically) are used depends upon user control of the solution procedure. This control is achieved by what body force file names are put in the input file. For the

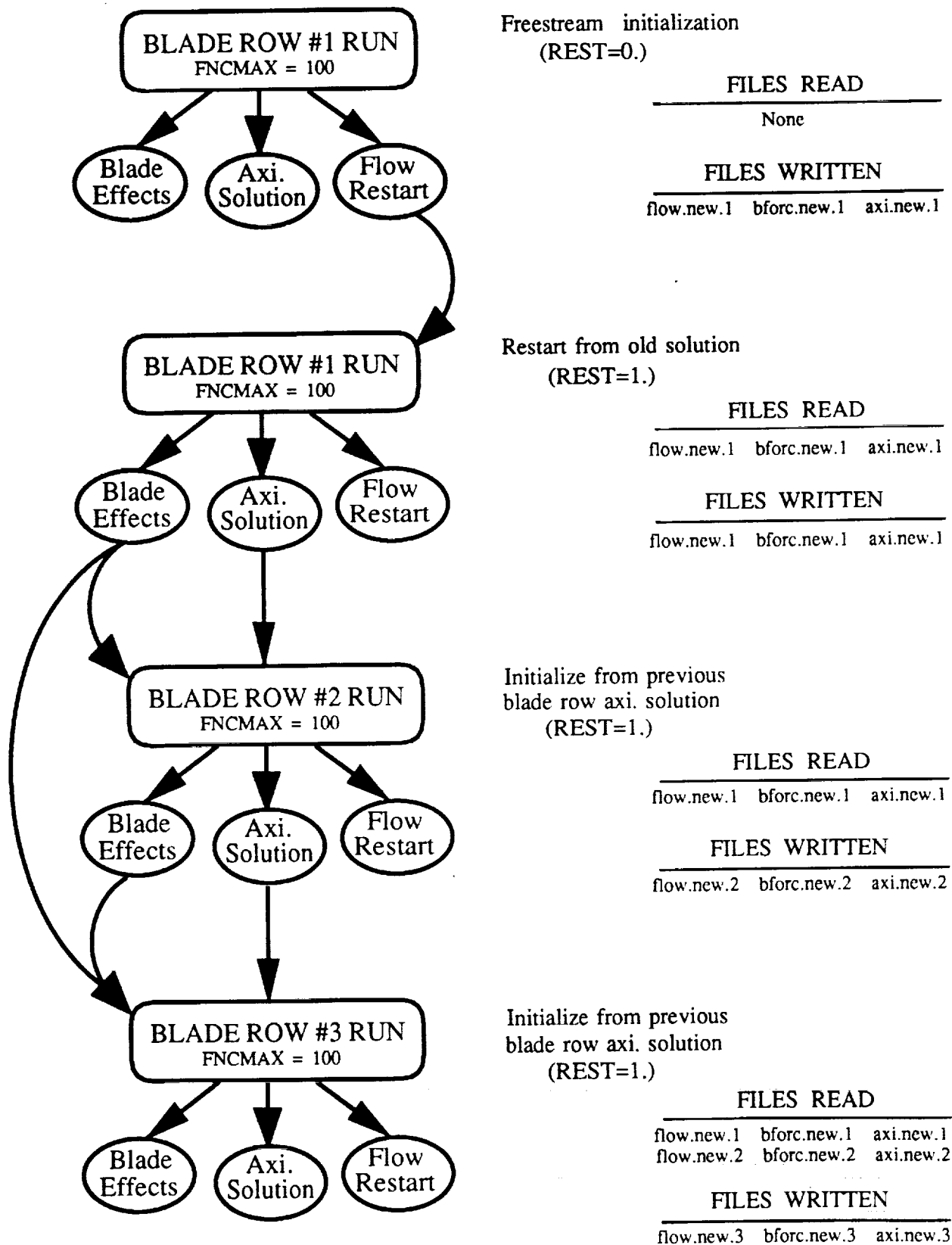


Figure 5.2: Diagram of a three blade row solution initialization

example of an N blade row calculation (see Fig. 5.3), the first blade row's input file could have a file name section as seen below.

flow.old.1	bforc.old.1	axi.old.1	grid.1
flow.old.2	bforc.old.2	axi.old.2	grid.2
.	.	.	.
.	.	.	.
flow.old.N	bforc.old.N	axi.old.N	grid.N
flow.new.1	bforc.new.1	axi.new.1	

The *bforc* files on the first N lines are read in and used in the calculation. The *bforc* name in the last line is the file where the updated blade effects for the first row are written. For the second blade row input file, the first line of the file name section could look like:

flow.old.1	bforc.old.1	axi.old.1	grid.1
------------	-------------	-----------	--------

which would direct *APES* to use the first blade row's body forces from the previous flip. To use the updated blade row effects, the first line should instead look like:

flow.new.1	bforc.new.1	axi.new.1	grid.1
------------	-------------	-----------	--------

For Fig. 5.3 the neighboring blade row effects are all from the previous flip, and the file names in the figure reflect that. However, if the updated terms for the current flip are used, the solution procedure is represented in 5.4.

This nested loop procedure is continued until the convergence criteria is met. Typically, a solution is deemed converged when the root mean square average of the density residual has been reduced by a factor of 10^{-3} from its initial value. For multiple blade row solutions, each flip is usually marked by an initial increase in the

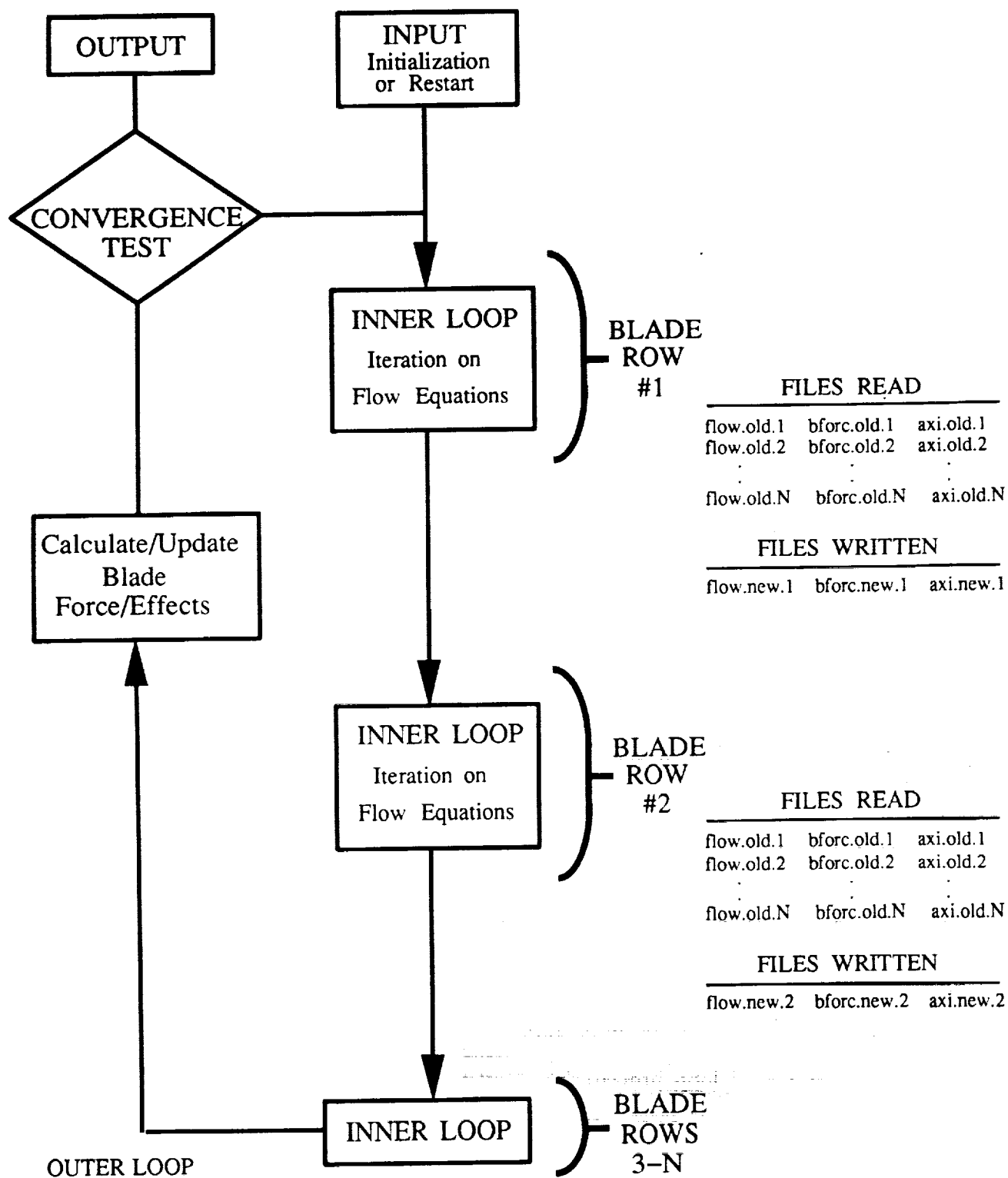


Figure 5.3: Multiple blade row solution procedure

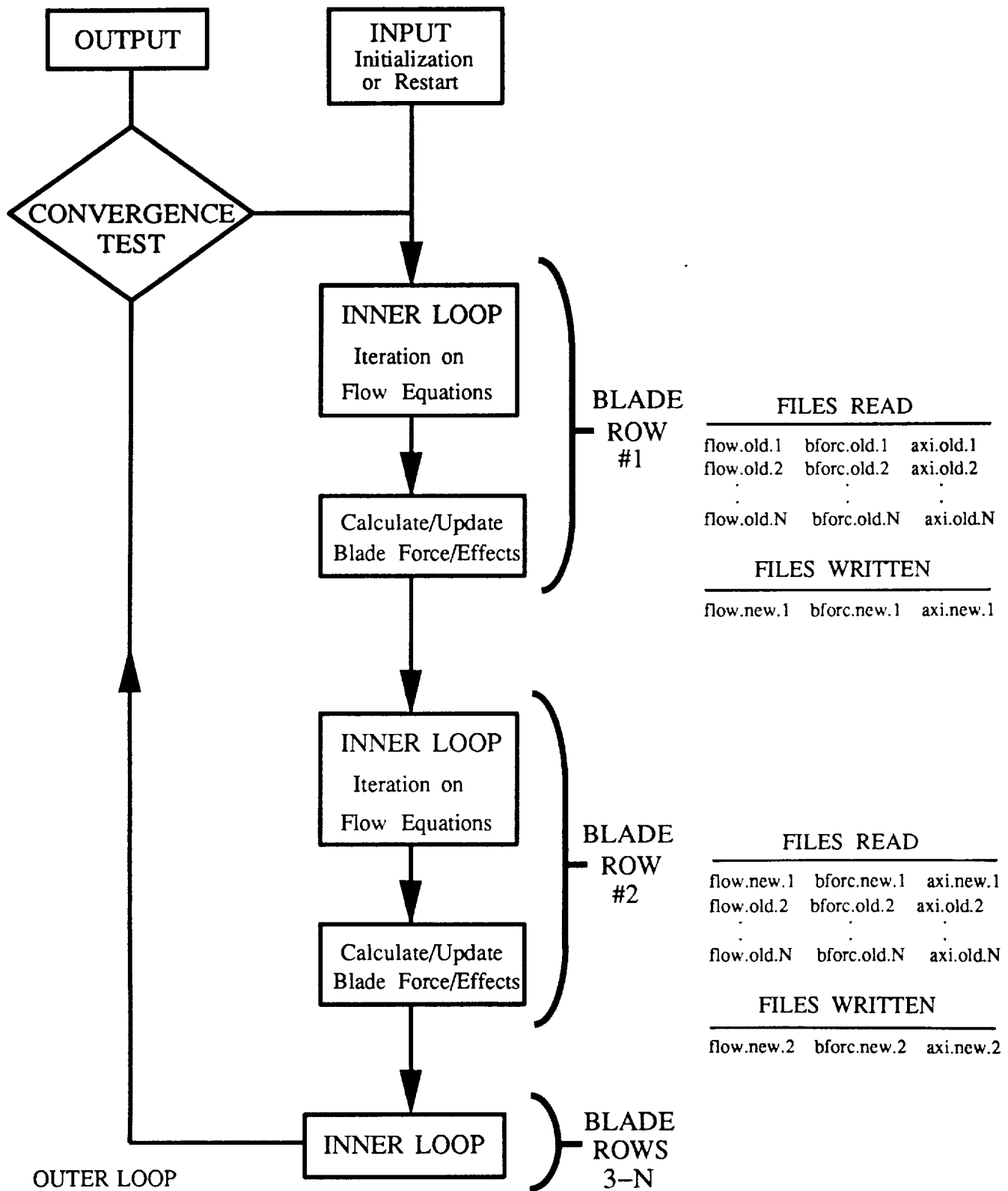


Figure 5.4: Solution procedure using updated blade row effects

residual, then a decrease so that a net reduction in the residual occurs. A second criterion (or measure of the trust worthness of the solution(s)) is that the mass flow at the inlet is close to the value at the exit. Typically, the difference between mass flows should be less than one percent of the inlet value. For multiple blade row solutions this criterion also applies to differences in mass flow between different blade row solutions.

There is no well defined requirement for the number of flips needed before a solution is deemed converged. Many factors contribute (e.g., number of blade rows, strength of aerodynamic coupling) to the number of flips necessary for convergence. At least two flips is recommended as a minimum number and there is no recommended maximum. A general guideline is when the individual solution's residuals have dropped to an acceptable level, run one more flip. If the mass flow, pressure ratio, or RMS average of the residuals do not change significantly, the solution system should be converged.

There are also no absolute rules governing the number of iterations each blade row must go through between flips. Generally, the number of iterations scales with grid size, with a reasonable choice being on the order of the number of axial grid points in the solution. This number should be enough to let all primary perturbations run the course of the domain. A number much less than this might not allow this "sonic communication" to occur properly. Also a number much more than this can unnecessarily settle a solution before the whole solution system (all other rows) is at a steady state. During a solution startup (covered in the next section) or other significant transients, a large number of iterations on each blade row can sometimes

bring about unrealistic and unstable aerodynamic situations.

For a single blade row case, the solution procedure is greatly simplified. With no neighboring blade rows and their effects to calculate, there is no outer loop. The iteration for the flow equation solver is executed until the convergence criteria is met.

5.2 Solution Startup Techniques

A solution startup procedure has been described for single and multiple blade rows in the previous section. The mechanics of the flow solver (*APES*) were covered but nothing was mentioned about overcoming some of the difficulties with the fluid mechanics of a solution startup. This section suggests startup techniques first for single blade row, and then multiple blade row geometries.

For a single blade row geometry, often a performance or flow condition is trying to be simulated. A match of rotational speed, mass flow, freestream Mach number, total pressure ratio, etc. is required. Setting the boundary conditions and other parameters to their final desired values and starting from freestream conditions might yield a converged, appropriate solution. However, with transonic fan sections, this often does not work. Starting up from freestream to an intermediate aerodynamic condition often yields good results. Two techniques are suggested and outlined below.

Often getting the shock system "swallowed" in the supersonic regions of a blade row (see Fig. 5.5) is desired. One way to achieve this is to set the exit static pressure to a low value (below the inlet total pressure value) as an initial condition. This induces the flow to go in the right direction and results in the flowfield being in a choked

(maximum mass flow) condition. From that point, the exit pressure is increased to the final desired value. For the example of an isolated shrouded rotor, this procedure might look like path *A* in Fig. 5.6 on a performance map.

A second procedure for starting up a transonic blade row is for the intermediate condition to be at a lower rotational speed than the final desired value. The rotational speed can be set to a fraction of the final value (.5 to .9) with the exit pressure set to a value appropriate to that condition. Once the solution is stabilized at this condition, the exit pressure and rotational speed are changed to their final values. This procedure is represented as path *B* on the performance map of Fig. 5.6. This procedure could eliminate the strong shock waves that occur with a full speed startup. These waves sometimes produce undesirable flow conditions and solution instabilities.

For multiple blade row solutions, the same procedures for a single blade row are used. An intermediate condition is often desirable, but sometimes flow conditions require a modified approach. For example, it is sometimes helpful to change the solution initialization process. The multiple blade row initialization procedure outlined in the previous section showed a continuous, serial progression through the rows. Another approach which has been successful is to initially make the outer loop iteration (see Fig. 5.3) between the first two blade rows alone. Once the two blade row system is stabilized, then the next blade row is initialized and included in the outer loop. This is repeated until all the blade rows are in the calculation.

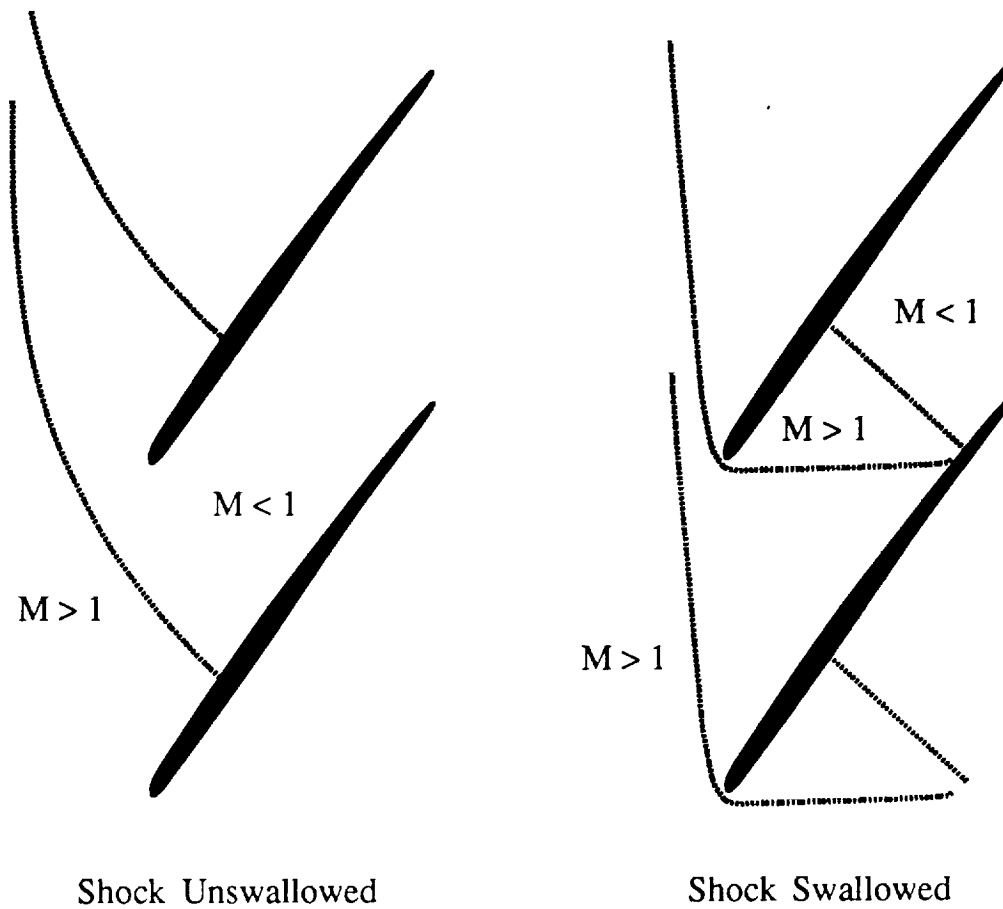


Figure 5.5: Shock system in a supersonic blade section

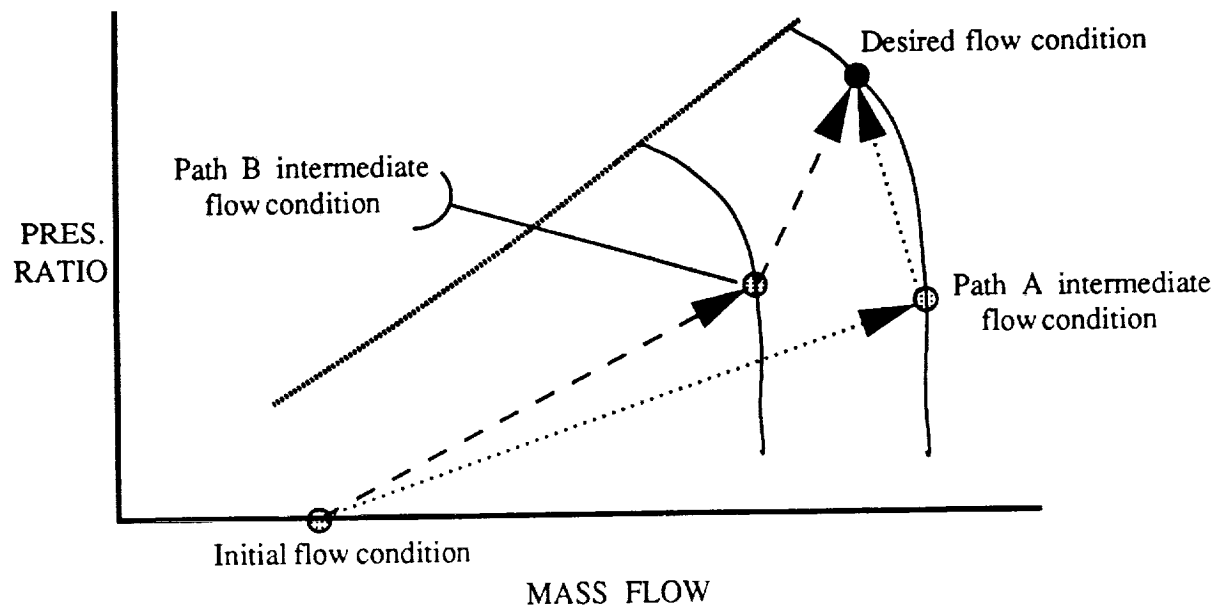


Figure 5.6: Generic performance map for an isolated rotor

REFERENCES

- [1] Celestina, M. L., Mulac, R. A., and Adamczyk, J. J., "A Numerical Simulation of the Inviscid Flow Through a Counterrotating Propeller", ASME Paper 86-GT-138, June 1986.
- [2] Whipple, D., "BDX-Binary Data Exchange Preliminary Information", NASA-Lewis Research Center, 1989.
- [3] Crook, A. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task IV - Advanced Fan Section Aerodynamic Analysis, Final Report", NASA CR 187128, NASA Contract NAS3-25270, 1992.
- [4] Hall, E. J., Delaney, R. A., and Bettner, J. L., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task II - Unsteady Ducted Propfan Analysis, Computer Program User's Manual", NASA CR 187105, NASA Contract NAS3-25270, 1991.
- [5] Crook, A. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task III - Advanced Fan Section Grid Generator, Final Report and Computer Program User's Manual", NASA CR 187129, NASA Contract NAS3-25270, 1991.

APPENDIX A. *FULLPLOT*: *PostScript* X-Y Plotting Routine

The *FULLPLOT* program was developed to provide an automated plotting facility for simple x-y type plots. The *ADPAC-APES* program produces files which monitor convergence history. These files are written in a format which *FULLPLOT* will use to create *PostScript* based plots. The execution of this program is extremely simple and can be invoked by the command:

fullplot <filename

where *filename* a plot file from an *APES* run.

Array limits in the *FULLPLOT* program are determined by the **PARAMETER** statement:

PARAMETER(MAXCUR=50, MAXPTS=5000)

The parameter **MAXCUR** determines the maximum number of curves that *FULLPLOT* can plot, while **MAXPTS** determines the maximum number of points per curve.

Input to *FULLPLOT* is directed upon execution, and this program can therefore easily be used for other two-dimensional plotting purposes. *FULLPLOT* produces a

PostScript plot file labelled *fort.15*, which can be sent directly to a *PostScript* printer, or previewed on a compatible device. The *FULLPLOT* input file may be modified by the user to construct the final plots in a number of forms. A sample *FULLPLOT* input file is given in Fig. A.1. The *PostScript* plot for this input file is given in Fig. A.2. Each of the various line and symbol types, as well as the *FULLPLOT* grayscale shading are illustrated in the sample plot.

The actual variables in the input file are free format. A description of each of the input PARAMETERS is given in the paragraphs below.

VARIABLE DESCRIPTION

TITLE Three lines of 80 character title data follow the TITLE header line. The 3 strings will be centered above the plot, and are plotted in the same order as they are input. Each title string must be in quotes.

LVH Vertical/horizontal plot trigger:

if = 1, the plot y axis is vertical on a standard $8\frac{1}{2}$ x 11 page (portrait mode),

if = 2, the plot y axis is horizontal on a standard $8\frac{1}{2}$ x 11 page (landscape mode).

LEGTYP Legend type plot trigger:

if = 0, no legend is plotted,

if = 1, a plain legend is plotted,

if = 2, a shadow box legend is plotted,

```

TITLE---- 3 LINES MAXIMUM -----
'This is the first title line'
'This is the second title line'
'This is the third title line'
LVH --- VERTICAL (0) OR HORIZONTAL (1) PLOT -----
0
LEGTFP---DXLEGI---DYLEGI---DXBOXI---(DXLEGI, DYLEGI INCHES FROM ORIGIN)---
2      0.25      7.0      3.5
LOGO-----DXLOGOI---DYLOGOI----- (LOGO =0 NO LOGO, DX, DY IN INCHES)---
1      -1.3      -1.5
XLABEL-----
'X axis Label'
NINCX-----XSMIN-----XSMAX-----LOCXAX-----NXSIG-----
5      0.0      1.0      0      2
YLABEL-----
'Y Axis Label'
NINCY-----YSMIN-----YSMAX-----LOCYAX-----NYSIG-----
5      -1.0      3.0      0      2
NUMCUR-----
14
CURVE DATA ----#PTS,LTYPE,STYPE,LEGEND,GRAYPLT,GRAYSCALE,LEGLABEL-----
2 1 1 1 0 0.0 'Line 1 Symbol 1'
0.1 0.1
0.1 0.9
2 2 2 1 0 0.0 'Line 2 Symbol 2'
0.2 0.1
0.2 0.9
2 3 3 1 0 0.0 'Line 3 Symbol 3'
0.3 0.1
0.3 0.9
2 4 4 1 0 0.0 'Line 4 Symbol 4'
0.4 0.1
0.4 0.9
2 5 5 1 0 0.0 'Line 5 Symbol 5'
0.5 0.1
0.5 0.9
2 6 6 1 0 0.0 'Line 6 Symbol 6'
0.6 0.1
0.6 0.9
2 7 7 1 0 0.0 'Line 7 Symbol 7'
0.7 0.1
0.7 0.9
2 8 8 1 0 0.0 'Line 8 Symbol 8'
0.1 -0.1
0.1 -0.9
2 9 9 1 0 0.0 'Line 9 Symbol 9'
0.2 -0.1
0.2 -0.9
2 10 10 1 0 0.0 'Line 10 Symbol 10'
0.3 -0.1
0.3 -0.9
2 11 11 1 0 0.0 'Line 11 Symbol 11'
0.4 -0.1
0.4 -0.9
2 12 12 1 0 0.0 'Line 12 Symbol 12'
0.5 -0.1
0.5 -0.9
2 13 13 1 0 0.0 'Line 13 Symbol 13'
0.6 -0.1
0.6 -0.9
5 1 0 1 1 0.5 'Gray Shaded - Grayscale=0.5'
0.8 -0.1
0.9 -0.5
0.8 -0.9
0.7 -0.5
0.8 -0.1

```

Figure A.1: Sample input data file for *FULLPLOT* plotting program

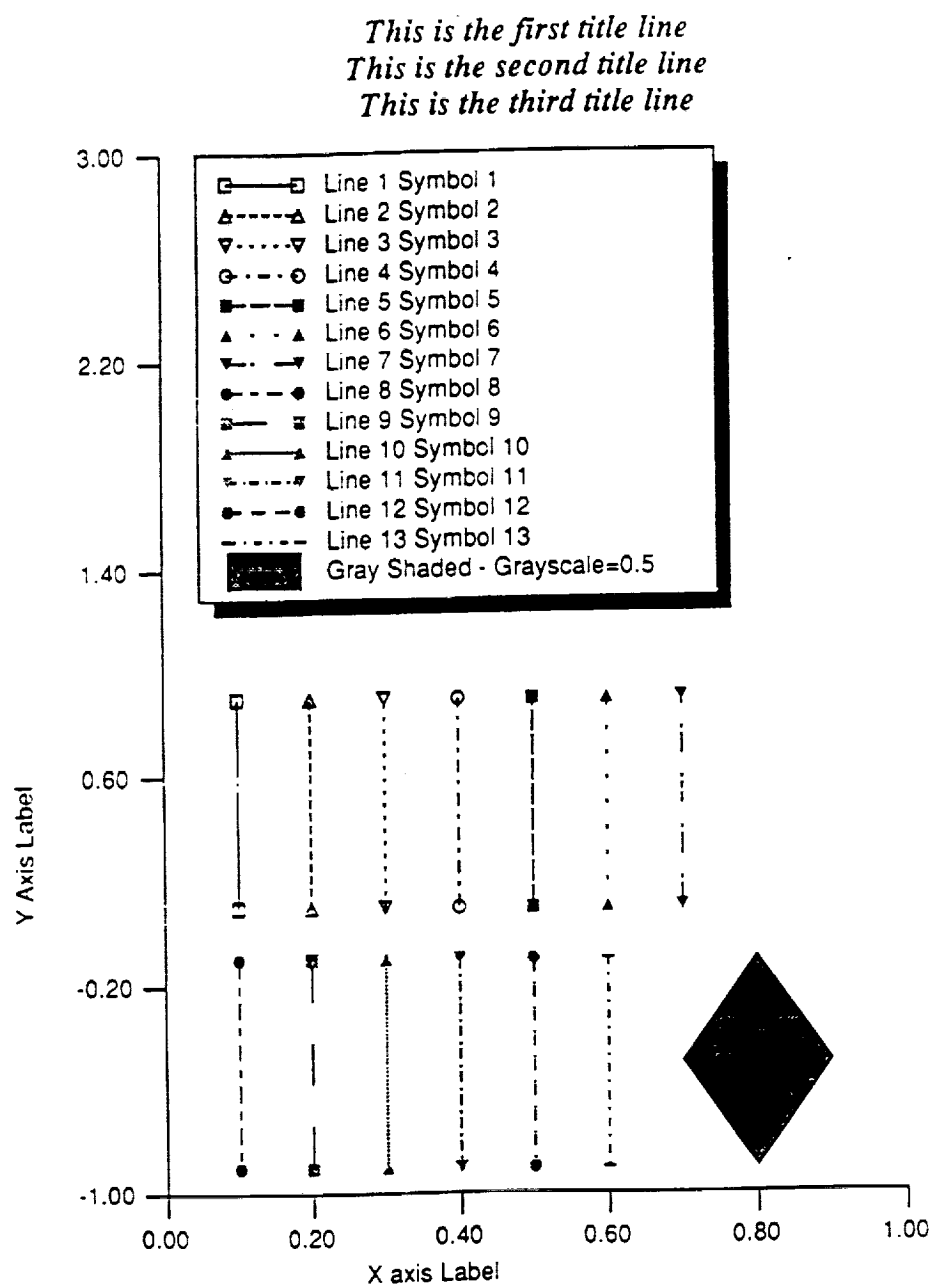


Figure A.2: Sample PostScript x-y plot from *FULLPLOT* plotting program

- DXLEGI** Legend placement variable. This variable determines the actual physical length, in inches, from the lower left hand legend corner to the x axis.
- DYLEGI** Legend placement variable. This variable determines the actual physical length, in inches, from the lower left hand legend corner to the y axis.
- DXBOXI** Legend box length parameter. The legend height is predetermined, but the legend box length is not. This variable permits user control of this function. This variable controls the actual physical length of the legend box in inches.
- LOGO** Logo plotting trigger:
if = 0, no logo is plotted,
if = 1, the NASA logo is plotted.
- DXLOGOI** Logo placement variable. This variable determines the actual physical length, in inches, from the lower left hand logo corner to the x axis.
- DYLOGOI** Logo placement variable. This variable determines the actual physical length, in inches, from the lower left hand logo corner to the y axis.
- XLABEL** X axis label. This is an 80 character string which will be centered below the x axis. The string must be in quotes.
- NINCX** Number of scale increments along the x axis. The x scale is determined by the number of increments and the minimum and maximum x axis scale values XSMIN, and XSMAX. The scale increment is then simply $(XSMAX - XSMIN) / (NINCX - 1)$

XSMIN X axis minimum scale value (see NINCX).

XSMAX X axis maximum scale value (see NINCX).

LOCXAX Not used.

NXSIG Number of significant digits past the decimal point in x axis scale markers.

YLABEL Y axis label. This is an 80 character string which will be centered along the y axis. The string must be in quotes.

NINCY Number of scale increments along the y axis. The y scale is determined by the number of increments and the minimum and maximum y axis scale values YSMIN, and YSMAX. The scale increment is then simply $(YSMAX-YSMIN) / (NINCY-1)$

YSMIN X axis minimum scale value (see NINCX).

YSMAX X axis maximum scale value (see NINCX).

LOCYAX Not used.

NYSIG Number of significant digits past the decimal point in y axis scale markers.

NUMCUR Number of separate sets of data (curves) to be drawn. Each curve is represented by a set of x-y data points which can be represented by points, lines, or as a shaded region on the plot.

The remainder of the *FULLPLOT* data set is the actual data to be plotted. The following data information must be repeated for each separate curve to be plotted. Each set of data (curve) begins with a header line with the following information:

#PTS Number of x-y data pairs in the curve.

LTYPE Line type used to draw the curve. A curve may be represented by lines, symbols, shaded regions, or all of the above.

The line types are defined as:

LTYPE=0 is no line (symbol only),

LTYPE=1 is a solid line,

LTYPE=2 is a medium dashed line,

LTYPE=3 is a short dashed line,

LTYPE=4 is a medium chained dash line,

LTYPE=5 is a very long dashed line.

LTYPE=6 is a very short dashed line.

LTYPE=7 is a staggered chain dashed line.

LTYPE=8 is a alternate dash length line.

LTYPE=9 is a very wide spaced dashed line.

LTYPE=10 is a very short spaced short dashed line.

LTYPE=11 is a tight staggered chained dashed line.

LTYPE=12 is a very tight staggered chained dashed line.

LTYPE=13 is a very short dash chained line.

STYPE Symbol type used to draw the curve. A curve may be represented by lines, symbols, shaded regions, or all of the above.

The symbol types are defined as:

STYPE=0 is no symbol (line only),

STYPE=1 is an open square,

STYPE=2 is an open triangle pointing up,

STYLE=3 is an open triangle pointing down,
STYLE=4 is an open circle,
STYLE=5 is an black filled square,
STYLE=6 is an black filled triangle pointing up,
STYLE=7 is an black filled triangle pointing down,
STYLE=8 is an black filled circle,
STYLE=9 is an gray filled square,
STYLE=10 is an gray filled triangle pointing up,
STYLE=11 is an gray filled triangle pointing down,
STYLE=12 is an gray filled circle.
STYLE=13 is a horizontal bar.

LEGEND A trigger to indicate whether the current curve should be included in the legend: if = 0, don't include this curve in the legend, if = 1, include this curve in the legend.

GRAYPLT A trigger to indicate whether the current curve should be plotted as a shaded region. If this is desired, the data will be interpreted as a closed curve which will have a shaded interior based on the shading value GRASCALE:

if = 0, don't treat this curve as a shaded region,
if = 1, treat this curve as a shaded region.

GRASCALE Shaded region shading value. This value must lie between 0.0 and 1.0. The value 0.0 represents true black, while a value of 1.0 represents true white. All other values in between represent increasing lighter shades

of gray. This value is only used when GRAYPLT=1.

LEGLABEL A character string (must be in quotes) indicating the legend label for this curve.

The remaining lines represent the #PTS pairs of x,y data in free format. One data set is given per line for the given curve.

APPENDIX 'B'

ADPAC-APES DISTRIBUTION LIST
User's Manual Cr-187127
Final Report Cr-187128
Contract NAS3-25270
Task Order #4

GOVERNMENT AGENCIES:

NASA Headquarters
600 Independence Avenue, SW
Washington, DC 20546
ATTN: RJ/R. Whitehead
RP/N. Nijahawan

NASA Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

ATTN: J.J. Adamczyk	M.S. 5-9 (2 copies)
C.L. Ball	M.S. 86-1
L.J. Bober	M.S. 77-6
D.R. Boldman	M.S. 86-7
B. Clark	M.S. 77-6
R.W. Claus	M.S. 142-5
J.H. Dittmar	M.S. 77-6
J.F. Groeneweg	M.S. 77-6 (4 copies)
C.E. Hughes	M.S. 77-6
R.J. Jeracki	M.S. 77-6
C.M. Kim	M.S. 77-6
A.P. Kurkov	M.S. 23-3
J. Lytle	M.S. ACC-1
C.J. Miller	M.S. 77-6 (15 copies)
D.P. Miller	M.S. 77-6
R.D. Moore	M.S. 77-6
D.V. Murthy	M.S. 23-3
L.D. Nichols	M.S. 142-5
C.W. Putt	M.S. 142-2 (2 copies)
D.R. Reddy	M.S. 5-11
T.S. Reddy	M.S. 23-3
R. Srivastava	M.S. 23-3
G.L. Stefko	M.S. 23-3
R.P. Woodward	M.S. 77-6
J.A. Ziemianski	M.S. 86-1
Report Control Office	M.S. 60-1 (4 copies)
Tech. Utilization Office	M.S. 7-3
AFSC Liaison Office	M.S. 501-3

NASA Ames Research Center
Moffett Field, CA 94035
ATTN: Library

M.S. 203-3

NASA Langley Research Center
Hampton, VA 23681-0001
ATTN: F. Farassat
S.L. Hodge
D. Stephens
Library

M.S. 460
M.S. 460
M.S. 462
M.S. 185

NASA Scientific & Technical
Information Facility
P.O. Box 8757
BWI Airport, MD 21240
ATTN: Accession Dept.

6 copies & FF427 form

Jason Assoc @ DARPA SUBTECH Center
David Taylor NSRDC
Bethesda, MD 20084
ATTN: C. Knight

Bldg. 17E, Room 120

Lockheed Engineering
and Sciences Company
144 Research Drive
Hampton, VA 23666
ATTN: M.H. Dunn

M.S. 904

Sverdrup Technology, Inc.
2001 Aerospace Parkway
Brookpark, OH 44142
ATTN: B. Berkowitz
R.M. Nallasamy
O. Yamamoto

SVR
SVR
SVR

INDUSTRY

AIRResearch Los Angeles Division
Allied-Signal Aerospace Company
P.O. Box 2960
Torrance, CA 90509-2960
ATTN: T. Booth

Dept. 93-209, T-42

Allison Gas Turbine Division, GMC Corp.

P.O. Box 420

Indianapolis, IN 46206-0420

ATTN: P.C. Tramm
R.F. Alverson
R.A. Delaney
D.W. Burns
J.F. Rathman
D.J. Helton
A.J. Crook
E.J. Hall
K.P. Nardini
J.L. Hansen
Library

M.S. T-11
M.S. T-14
M.S. T-14A
M.S. T-10B
M.S. T-02A
M.S. T-14A
M.S. T-14A
M.S. T-14A
M.S. T-14A
M.S. S-50
M.S. S-05

Boeing Commercial Airplane Company

(BCAC)

P.O. Box 3707

Seattle, WA 98124-2207

ATTN: B. Farquhar
W.H. Jou
R. Cuthbertson
G. Sengupta

M.S. 6M-98
M.S. 7H-96
M.S. 79-84
M.S. 7H-91

Douglas Aircraft Company Division

McDonnell Douglas Corporation

3855 Lakewood Boulevard

Long Beach, CA 90846

ATTN: M. Joshi
F. Lynch
G. Page
W. Siegele

M.S. 36-60
M.S. 36-60
M.S. 35-86
M.S. 202-15

General Dynamics Convair

P.O. Box 80844

San Diego, CA 92138

ATTN: B. Bergman
K. Taylor

M.Z. 36-1240
M.Z. 54-6890

General Electric Company

Aircraft Engine Group

1 Neumann Way

Evendale, OH 45215

ATTN: P. Gliebe
C. Lenhardt
M. Majjigi
M. Pearson
L. Smith
C. Whitfield

Mail Drop A-304
Mail Drop A-330
Mail Drop A-319
Mail Drop A-317
Mail Drop H-4
Mail Drop A-304

Hamilton Standard Division-UTC
Windsor Locks, CT 06096
ATTN: F.B. Metzger

M.S. 1A-3-6

Lockheed California Company
P.O. Box 551
Burbank, CA 91520
ATTN: Library

Pratt & Whitney Aircraft-UTC
Commercial Products Division
400 Main Street
East Hartford, CT 06108

ATTN: D.B. Hanson
D. Hopwood
D. Mathews
W. Lord
T. Wynosky

M.S. 165-11
M.S. 162-07 (3 copies)
M.S. 165-11
M.S. 169-23
M.S. 169-23

Rohr Industries, Inc.
P.O. Box 878
Chula Vista, CA 92012-0878
ATTN: Library

United Technologies Corporation
Research Center
Silver Lane
East Hartford, CT 06108

ATTN: M. Barnett
R. Davis
D. Dorney

M.S. 20
M.S. 20
M.S. 20

UNIVERSITIES

Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332-0800
ATTN: Dr. L.N. Sankar

Mississippi State University
Department of Aerospace Engineering
P.O. Box A
Mississippi State, MS 39762
ATTN: Dr. D. Whitfield
Dr. M. Janus

Ohio State University
Department of Aeronautical
and Astronautical Engineering
Columbus, OH 43220
ATTN: Prof. G.M. Gregorek

Pennsylvania State University
Department of Aerospace Engineering
233 Hammond Building
ATTN: Dr. B. Lakshminarayana

Purdue University
School of Aeronautics and Astronautics
West Lafayette, IN 47907
ATTN: Dr. J.P. Sullivan
Dr. M. Williams

Purdue University
School of Mechanical Engineering
West Lafayette, IN 47907
ATTN: Dr. S. Fleeter

Texas A&M University
Aerospace Engineering Department
College Station, TX 77843-3141
ATTN: Dr. K.D. Korkan

University of Arizona
Aerospace and Mechanical Engineering
Department
Aero Building #16
Tucson, AZ 85721
ATTN: Dr.E.J. Kerschen

University of Iowa
Institute of Hydraulic Research
Iowa City, IA 52242
ATTN: F. Stern

University of Missouri-Rolla
Mechanical Engineering Department
Rolla, MO 65401-0249
ATTN: Dr. Walter Eversman

1301nj



Report Documentation Page

1. Report No. NASA-CR-187127	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle INVESTIGATION OF ADVANCED COUNTERROTATION BLADE CONFIGURATION CONCEPTS FOR HIGH SPEED TURBOPROP SYSTEMS, TASK IV-ADVANCED FAN SECTION AERODYNAMIC ANALYSIS COMPUTER PROGRAM USER'S MANUAL.		5. Report Date NOVEMBER, 1992
		6. Performing Organization Code
7. Author(s) ANDREW J. CROOK ROBERT A. DELANEY		8. Performing Organization Report No.
		10. Work Unit No. 535-03-10
9. Performing Organization Name and Address ALLISON GAS TURBINE DIVISION OF GENERAL MOTORS CORP, P.O. BOX 420, SC T14A INDIANAPOLIS, IN 46206-0420		11. Contract or Grant No. NAS3-25270
		13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS & SPACE ADMINISTRATION LEWIS RESEARCH CENTER 21000 BROOKPARK ROAD CLEVELAND, OH 44135-3191		14. Sponsoring Agency Code
15. Supplementary Notes PREPARED IN COOPERATION WITH NASA PROJECT MANAGER, C.J. MILLER, NASA LEWIS RESEARCH CENTER, CLEVELAND, OHIO		
16. Abstract THIS DOCUMENT CONTAINS THE COMPUTER PROGRAM USER'S MANUAL FOR THE ADPAC- APES (ADVANCED DUCTED PROPFAN ANALYSIS CODES-AVERAGE PASSAGE ENGINE SIMULATION) PROGRAM DEVELOPED TASK IV OF NASA CONTRACT NAS3-25270. THE OBJECTIVE OF THIS TASK IS THE DEVELOPMENT OF A THREE-DIMENSIONAL EULER/NAVIER-STOKES FLOW ANALYSIS FOR FAN SECTION/ENGINE GEOMETRIES CONTAINING MULTIPLE BLADE ROWS AND MULTIPLE SPANWISE FLOW SPLITTERS. AN EXISTING PROCEDURE DEVELOPED BY DR. J. J. ADAMCZYK AND ASSOCIATES AT THE NASA LEWIS RESEARCH CENTER WAS MODIFIED TO ACCEPT MULTIPLE SPANWISE SPLITTER GEOMETRIES AND SIMULATE ENGINE CORE CONDITIONS. THE NUMERICAL SOLUTION IS BASED UPON A FINITE VOLUME TECHNIQUE WITH A FOUR STAGE RUNGE-KUTTA TIME MARCHING PROCEDURE. MULTIPLE BLADE ROW SOLUTIONS ARE BASED UPON THE AVERAGE-PASSAGE SYSTEM OF EQUATIONS. THE NUMERICAL SOLUTIONS ARE PERFORMED ON AN H-TYPE GRID SYSTEM, WITH MESHES MEETING THE REQUIREMENT OF MAINTAINING A COMMON AXISYMMETRIC MESH FOR EACH BLADE ROW GRID. THE ANALYSIS WAS RUN ON SEVERAL GEOMETRY CONFIGURATIONS RANGING FROM ONE TO FIVE BLADE ROWS AND FROM ONE TO FOUR RADIAL FLOW SPLITTERS. THE EFFICIENCY OF THE SOLUTION PROCEDURE WAS SHOWN TO BE THE SAME AS THE ORIGINAL ANALYSIS.		
17. Key Words (Suggested by Author(s)) EULER, NAVIER-STOKES, ADPAC, APES, NPSS, TURBOFAN, PROPFAN, DUCTED		18. Distribution Statement UNCLASSIFIED-UNLIMITED
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of pages 75
		22. Price